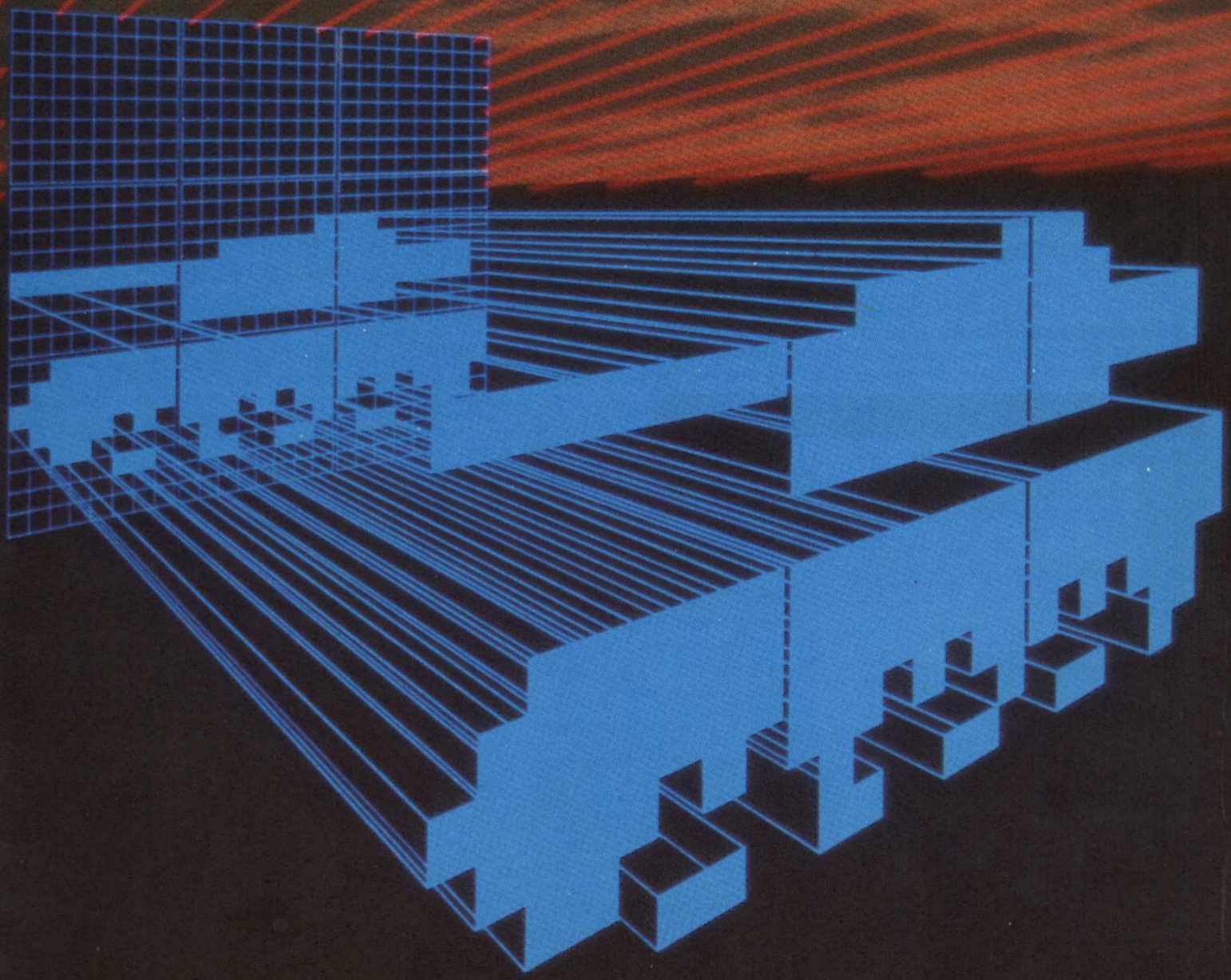
CURSO PRÁTICO 29 DE PROGRAMAÇÃO DE COMPUTADORES

INPUT

PROGRAMAÇÃO BASIC - PROGRAMAÇÃO DE JOGOS - CÓDIGO DE MÁQUINA

NOVA CULTURAL

Cz$ 55,00



INPUT

Vol. 2 Nº 29

NESTE NÚMERO

PERIFÉRICOS

SUA LIGAÇÃO COM O MUNDO

Faça o seu micro comunicar-se com outros computadores, utilizando o modem como elo de ligação entre ele e o mundo 561

PROGRAMAÇÃO BASIC

CONJUNTOS DE BLOCOS GRÁFICOS (3)

De como um macaco e uma cobra podem ajudá-lo a trabalhar com blocos gráficos. Linhas **DATA** que complementam o cenário 565

CÓDIGO DE MÁQUINA

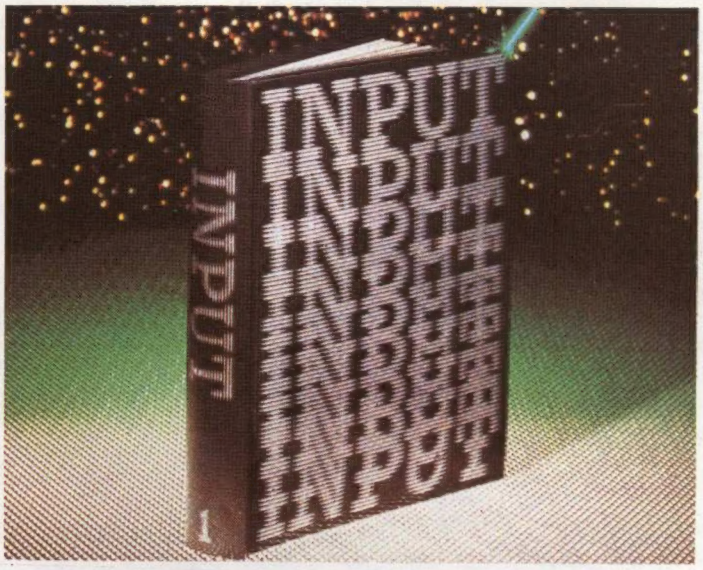
COMO FUNCIONA O GERADOR GRÁFICO

Como imprimir e apagar. Cálculo das posições de impressão. Contadores 570

APLICAÇÕES

UM EDITOR DE TEXTOS (1)

Analise o menu. Nomes de arquivos. Parâmetros de entrada e saída. O programa editor de textos. Ajuste as cores 576



PLANO DA OBRA
INPUT é uma obra editada em fascículos semanais, e cada conjunto de 15 fascículos compõe um volume. A capa para encadernação de cada volume estará à venda oportunamente.

FÉRIAS, VIAGENS, MUDANÇAS...
NÃO FIQUE COM A COLEÇÃO INCOMPLETA
Se você está saindo de férias, pretende viajar ou vai se ausentar por algum tempo, avise antecipadamente seu jornaleiro. Ele pode guardar os seus fascículos enquanto você estiver fora. Se, por qualquer motivo, você perdeu alguns números, peça-os também a seu jornaleiro, ou entre em contato com nossa Distribuidora:

1. **Pessoalmente** — Em *São Paulo,* os endereços são: rua Brigadeiro Tobias, 773, Centro; av. Industrial, 117, Santo André. No *Rio de Janeiro,* av. Mem de Sá, 191/193, Centro.
2. **Por carta** — Envie para:
   DINAP — Distribuidora Nacional de Publicações
   Números Atrasados
   Estrada Velha de Osasco, 132 — Jardim Teresa
   CEP 06040 — Osasco — SP
3. **Por telex** — Utilize o nº (011) 33 670 DNAP.

Em **Portugal,** os pedidos devem ser feitos à Distribuidora Jardim de Publicações Lda. — Qta. Pau Varais, Azinhaga de Fetais, 2685, Camarate, Lisboa; Apartado 57; Telex 43 069 JARLIS P.
Atenção: Após seis meses do encerramento da coleção, o atendimento dos pedidos dependerá da disponibilidade do estoque.
Obs.: Quando pedir livros, mencione sempre o título e/ou autor da obra, além do número da edição.

COLABORE CONOSCO
Encaminhe seus comentários, críticas, sugestões ou reclamações ao
SERVIÇO DE ATENDIMENTO AO LEITOR
Caixa Postal 9 442, São Paulo — SP.

EDITOR
RICHARD CIVITA
NOVA CULTURAL

Presidente
Flávio Barros Pinto
Diretoria
Carmo Chagas, Iara Rodrigues,
Pierluigi Bracco, Plácido Nicoletto,
Roberto Silveira, Shozi Ikeda,
Sônia Carvalho

REDAÇÃO
Diretor Editorial: Carmo Chagas
Editores Executivos:
Antonio José Filho, Berta Sztark Amar
Editor Chefe: Paulo de Almeida
Editoras Assistentes: Ana Lúcia B. de Lucena,
Marisa Soares de Andrade
Chefe de Arte: Carlos Luiz Batista
Assistentes de Arte: Dagmar Bastos Sampaio,
Grace Alonso Arruda, Monica Lenardon Corradi

Secretária de Redação/Coordenadora: Stefania Crema
Secretário de Redação: Mauro de Queiroz

Colaboradores
Consultor Editorial Responsável:
Dr. Renato M. E. Sabbatini
(Diretor do Núcleo de Informática Biomédica da
Universidade Estadual de Campinas-SP)
Execução Editorial: DATAQUEST Assessoria
em Informática Ltda., Campinas
Tradução, adaptação, programação e redação:
Abílio Pedro Neto, Aluísio J. Dornellas de Barros,
Marcelo R. Pires Therezo, Marcos Huascar Velasco,
Raul Neder Porrelli, Ricardo J. P. de Aquino Pereira
Coordenação Geral: Rejane Felizatti Sabbatini

COMERCIAL
Diretor Comercial: Roberto Silveira
Gerente Comercial: Joaquim Celestino da Silva
Gerente de Circulação: Denise Mozol
Gerente de Propaganda e Publicidade: José Carlos Madio
Gerente de Pesquisa e Análise de Mercado:
Wagner M. P. Nabuco de Araújo

CLC

A Editora Nova Cultural Ltda. é uma empresa do
Grupo CLC — Comunicações, Lazer, Cultura S.A.
Presidente: Richard Civita
Diretoria: Flávio Barros Pinto, João Gomez,
Menahen M. Politi, Renê C. X. Santos,
Stélio Alves Campos

© Marshall Cavendish Limited, 1984/85.
© Editora Nova Cultural Ltda., São Paulo,
Brasil, 1986; 2ª edição, 1987.

Edição organizada pela Editora Nova Cultural Ltda.
Av. Brigadeiro Faria Lima, 2000 - 3º andar
CEP 01452 - São Paulo - SP - Brasil
(Artigo 15 da Lei 5 988, de 14/12/1973).
Esta obra foi composta pela AM Produções Gráficas Ltda.
e impressa pela Companhia Lithographica Ypiranga.

SUA LIGAÇÃO COM O MUNDO

- CONEXÃO COM O MUNDO
- COMPRAS EM CASA
- TELETEXTO E VIDEOTEXTO
- LIGAÇÃO EM REDE
- TIPOS DE MODEM

Se você deseja explorar plenamente o potencial de seu micro, faça-o comunicar-se com outros computadores, utilizando o modem como elo de ligação entre ele e o mundo.

Dos computadores pode-se dizer que, como homens, nenhum deles é uma ilha... Pois até mesmo os mais modestos dentre eles podem ser conectados ao sistema telefônico e, em conseqüência, a alguns dos mais poderosos "cérebros eletrônicos" e bancos de dados da Terra. Ligado desse modo, um micro nos proporciona acesso a quantidades inesgotáveis de informação.

Um computador, um dispositivo de ligação chamado *modem*, um telefone e um software muito simples: eis tudo o que é preciso para realizar esse milagre da informática.

A comunicação entre computadores se dá de três modos básicos: por meio de linhas telefônicas ou de ondas de rádio (ou ambos, em alguns casos); através da co-participação no mesmo sistema de armazenamento de informação (que, na prática, significa co-participação na mesma unidade de disco); ou com a ajuda de um simples cabo que conecte dois aparelhos — neste caso, cada um deles pode compartilhar o processador e a memória do outro.

ALÔ, ALÔ... MUNDO

Dessas opções, a primeira é sem dúvida a mais excitante, pois representa um passo à frente para quem é entusiasta dos computadores pessoais.

Atualmente, já é possível obter, a preços razoáveis, equipamento e software que farão com que o seu aparelho seja capaz de conversar com outro em qualquer lugar do globo, usando as mesmas linhas empregadas pelos sistemas de telecomunicações.

O sistema telefônico proporciona acesso a uma imensa faixa de facilidades, como o intercâmbio de software e a ampliação dos serviços de notícias e informações, que hoje custam muito caro, quando realizados pelos meios convencionais (tele-trabalho).

Com um computador e um modem, você pode consultar listas de compras, examinar ilustrações de produtos para venda, comparar preços, pagar contas instantaneamente, verificar de modo automático como andam as suas finanças, e fazer quase tudo o que quiser, sem sair de casa.

Para trabalhar juntos, uma secretária e seu chefe, por exemplo, não precisam estar na mesma sala, no mesmo edifício ou na mesma cidade. Basta para isso que usem formas de comunicação por computador.

Assim, tendo esboçado uma carta no micro, o chefe pode enviar todo o rascunho, com erros ortográficos e de gramática, à sua secretária, que o corrigirá, o formatará de modo a torná-lo apresentável e o enviará.

BOLETINS

Diversos sistemas telemétricos, como o videotexto e o Cirandão, permitem que o usuário estabeleça a sua própria "caixa postal". Assim, se ele quiser entrar em contato com outro usuário, pode "escrever" uma carta no processador de textos do micro, e guardá-la na "caixa postal" do destinatário. Este último, por sua vez, poderá copiá-la em seu microcomputador, a fim de lê-la mais tarde. Certamente, tanto quem envia como quem recebe a carta deve ter um computador ligado à rede. Um *Correio Eletrônico* desse tipo constitui uma poderosa ferramenta, e representa um grande progresso em relação aos boletins.

REDES

Muitas das aplicações mais interessantes para a comunicação entre computadores envolvem pequenos aparelhos com acesso a um grande "cérebro eletrônico". Entretanto, outras combinações são possíveis.

Uma vida solitária? Talvez, mas não necessariamente. Também é possível contatar outras pessoas com os mesmos

interessses em computadores. Em alguns casos, independentemente do lugar do mundo em que estejam essas pessoas, o custo do contato com elas pode ser tão pequeno quanto uma simples ligação telefônica local. Você pode chamar aquilo que é conhecido por *boletim* (*bulletin boards*, em inglês, comumente traduzido para "quadro de avisos") para ler mensagens enviadas por outras pessoas e também deixar suas próprias mensagens. Existem boletins programados para vários propósitos.

Se você decidisse abandonar o conforto de seu lar e aventurar-se a sair pelo mundo afora, poderia deixar registrado quase tudo no computador — despesas com táxis e hotéis, dias de férias, entradas para teatro, passagens de avião, e muito mais.

Mesmo que surjam problemas nas ligações de computadores (causados sobretudo por falhas no estabelecimento de padrões comuns), quase sempre se dá um jeito; muitos desses trabalhos pioneiros são realizados por amadores e estudantes entusiasmados e seus professores em universidades e outros estabelecimentos de ensino superior.

O crescimento do interesse em comunicações entre computadores e as indicações de que esta será a área de exploração que mais se desenvolverá no futuro refletem-se claramente no fato de que mais e mais boletins computadorizados estão sendo criados em todo o mundo (inclusive no Brasil).

Embora possam ser feitos pelo próprio usuário, os boletins são, em sua maioria, montados por empresas, geralmente especializadas no ramo da eletrônica. Alguns desses "boletins empresariais" reservam um espaço para os usuários comuns, enquanto outros ocupam a área do quadro apenas para publicar seus serviços e benefícios. Ultimamente, porém, o setor de maior crescimento tem sido o dos boletins criados por amadores e universitários.

COMPRAS EM CASA

Ainda há quem imagine que essa aplicação da comunicação entre computadores pertence ao domínio da ficção científica, e que será necessário esperar muitos anos, até que seja possível realizar, por exemplo, compras por intermédio de um terminal de vídeo em casa. Entretanto, tudo isso já é possível atualmente. Nos países mais desenvolvidos, muitas pessoas utilizam esse sistema, e estão tão familiarizadas com ele, que as telecompras já fazem parte de sua rotina diária. No Brasil, o sistema videotexto oferece essa possibilidade em várias capitais.

Nos lugares em que isso já acontece, você pode usar o seu computador para ter acesso instantâneo a informações sobre produtos, ofertas especiais e preços em um grande número de lojas. Ou pode pedir serviços e pagá-los com a simples digitação de algumas teclas. O "trabalho" é feito por um computador central. Ele contém toda a informação e também faz a transferência eletrônica de dinheiro da conta do cliente no banco para a conta da loja.

TELETEXTO E VIDEOTEXTO

Igualmente crescente é o número de serviços de teletexto e videotexto, aos

quais é possível conectar até mesmo pequenos micros domésticos. O serviço de teletexto — que ainda não existe no Brasil — é o que sai mais barato para o usuário, pois não é interativo. Desenvolvido na Inglaterra, esse serviço segue dois padrões adotados em vários países: o ORACLE (executado pelas companhias de televisão independentes) e o CEEFAX (executado pela BBC). Com o teletexto, o usuário utiliza um televisor adaptado para receber em casa milhares de páginas de informação (textos e figuras, por exemplo), que são mostradas no vídeo. Não é necessário o telefone, pois as páginas são enviadas por radiodifusão, no intervalo entre uma tela e outra da programação normal de TV.

Enviar e carregar programas são duas funções de grande interesse em comunicação de computadores; elas podem ser desempenhadas tanto no videotexto quanto no teletexto. Programas podem ser transmitidos também por TV ou por rádio. Porém, os programas do teletexto têm uma vantagem: eles podem ser carregados diretamente no micro, enquanto os programas obtidos por meio do rádio precisam ser primeiramente gravados em fita para, só então, serem carregados pelo modo normal.

A única desvantagem do teletexto é que, como ele usa sinais de radiodifusão, a comunicação se dá apenas em um sentido. Já o videotexto (que emprega uma linha telefônica para estabelecer comunicação) é bidirecional. Isto significa que a sua máquina pode "conversar" com o computador central do videotexto. Na realidade, a rede de videotexto pode incluir vários computadores regionais interligados, todos oferecendo basicamente o mesmo serviço. Essa multiplicidade de computadores ajuda a manter baixo o preço do sistema para o usuário, que já paga taxas altas só para usar o telefone. Quanto mais perto o computador estiver do assinante, mais barata será a ligação.

Outro serviço público de telecomunicação entre computadores, bem mais diferenciado e sofisticado que o videotexto e o teletexto, é constituído pelas redes de microcomputadores, como por exemplo o sistema Cirandão, da empresa estatal Embratel. Esse sistema proporciona aos proprietários de micros a possibilidade de conversarem entre si, empregando uma rede de correio eletrônico ("caixas postais"). Além disso, ele oferece também muitos outros serviços de "boletim computadorizado", teleconferência, consulta a bancos de dados, e recepção e transmissão de programas de computador.

Uma desvantagem de sistemas com muitos usuários e que utilizam a rede telefônica normal, como o Cirandão, é a sua baixa velocidade de transmissão. Isso acontece porque as redes telefônicas brasileiras não suportam um ritmo muito intenso de transmissão, sem degradação do sinal. Por isso, usuários profissionais (bancos, por exemplo) são obrigados a recorrer a redes especiais de transmissão em alta velocidade, como é o caso de outro serviço da Embratel, o Transdata.

O Transdata permite enviar e receber informações em uma faixa maior de velocidade, o que significa também que computadores de diferentes tamanhos e velocidades podem compartilhar os serviços. Estes, porém, são muito mais caros do que os cobrados pelo Cirandão e pelo videotexto.

O acesso a bancos de dados e redes de computadores situados em outros

países é possível por intermédio do serviço Interdata da Embratel.

CORREIO ELETRÔNICO

Uma rede pode interligar um computador principal, microcomputadores e qualquer espécie de periféricos. O jornal norte-americano *New York Times*, por exemplo, usa os três tipos de computador em um dos mais sofisticados sistemas de produção de informações. Em sistemas dessa envergadura, muitas pessoas que trabalham em escritório podem usar o seu próprio terminal ligado a um minicomputador, enquanto outro terminal no escritório se comunica com um computador principal.

Desse modo, o trabalho do dia-a-dia, no caso de uma redação de jornal, pode ser preenchido por um simples minicomputador, enquanto o computador central é reservado para o uso da administração da empresa e para os bancos de dados principais.

Por outro lado, os jornalistas que estiverem trabalhando fora da redação podem valer-se de um microcomputador com um modem para ter acesso aos bancos de dados, visando obter informações para suas reportagens. Estas podem ser igualmente enviadas à redação por meio de um computador e um modem.

Em muitos países, é comum o fato de jornalistas ("cobrindo", por exemplo, uma partida de futebol ou uma corrida de Fórmula-1) utilizarem diretamente um teclado e um vídeo, em lugar de uma máquina de escrever. Ao serem apertadas algumas teclas, o resumo e o resultado do jogo são instantaneamente enviados ao computador da redação.

Como os fabricantes de computadores ainda não se ajustaram à necessidade de estabelecer padrões comuns internacionais, a comunicação entre computadores não é, por enquanto, tão fluida como se poderia desejar. Entretanto, os problemas que concorrem para isso estão sendo solucionados e, aos poucos, os computadores vão se tornando compatíveis entre si. Isso tem acontecido principalmente no caso da comunicação entre computadores por intermédio de linhas telefônicas.

SEGURANÇA

O fato de computadores diferentes poderem se comunicar uns com os outros através de linhas telefônicas agrada muitos possuidores de micros, mas assusta universidades e grandes empresas, bancos e agências do governo. O problema é que tais instituições também utilizam esse tipo de comunicação. Isto as torna vulneráveis pois, apesar das medidas de segurança adotadas pelas autoridades para garantir a confidencialidade das informações armazenadas e transmitidas, seus computadores são, vez por outra, "invadidos" por usuários não autorizados.

Esse é o tema do filme *Jogos de Guerra*, que foi objeto de muita controvérsia, especialmente nos Estados Unidos, onde a fraude em sistemas telemétricos tem causado prejuízos de milhões de dólares.

Os crimes eletrônicos são favorecidos pela relativa facilidade com que se processa a comunicação entre computadores. Em vez da violência física, os criminosos utilizam, nesse caso, métodos bem mais sutis e sofisticados. Um dos maiores problemas com que se defrontam as instituições fraudadas tem sido, aliás, saber se estão realmente sendo roubadas. Como muitos serviços são feitos por meio de linhas telefônicas, a investigação do crime eletrônico vai se tornando cada vez mais difícil, particularmente quando o próprio criminoso pode efetuar a mudança dos registros relativos ao crime.

OUTROS MEIOS DE COMUNICAÇÃO

Os telefones não são o único meio de comunicação entre computadores. Ondas de rádio também podem ser usadas, desde que haja dois computadores conectados a equipamentos capazes de transmiti-las e recebê-las.

As fibras ópticas são atualmente o meio pelo qual a comunicação entre computadores se processa de modo mais rápido. Nesse caso, a informação é codificada em impulsos de luz e transmitida por meio de finos fios de vidro. Essa nova tecnologia parece ser mais eficiente do que qualquer outro método conhecido: ela faz com que a informação viaje à velocidade da luz. As comunicações por fibra óptica já passaram da fase experimental, e estão sendo usadas em diversos países, inclusive no Brasil.

Os raios infravermelhos também vêm sendo testados como meio de comunicação entre micros numa mesma sala. A vantagem principal é que se elimina o "cipoal" de fios que acompanha toda grande instalação de computadores.

TORRE DE BABEL

Não importa o meio de comunicação sempre existirá problema de escolhido: compatibilidade. Isso acontece porque um micro só pode se comunicar com outro micro (ou com outro dispositivo periférico) por intermédio da interface correta. Sem a interligação correta de cabos e soquetes e um código que as duas máquinas entendam, a comunicação torna-se impossível.

A linha telefônica é usada para carregar informações de um computador para o outro através de longas distâncias. Entretanto, ela aceita apenas sinais seriais.

Isso não significa que os computadores com interfaces paralelas não sejam capazes de enviar sinais pela linha telefônica. Existem circuitos eletrônicos e programas disponíveis que tornaram possível essa emissão. Tais dispositivos, porém, podem sair mais caros do que o próprio computador.

Um dos problemas com o uso de telefone para comunicação entre computadores é que estes carregam informações em impulsos ou pulsos elétricos discretos e separados. Mas os telefones atuais são projetados para transmitir a voz humana — que é um sinal analógico, contínuo e variável. Assim, o dado de um computador precisa ser convertido em um sinal similar.

MODEMS

O equipamento usado para converter os dados do computador em sinais que possam ser transmitidos através das linhas telefônicas, e vice-versa, é o modem. A palavra é uma abreviação de MOdulador/DEModulador, que é a descrição da função do modem.

Existem dois tipos de modem, o acoplador acústico e o modem de conexão direta. Ambos variam de formas e tamanhos, mas estão geralmente contidos em uma caixa. O acoplador acústico tem duas alças de borracha para acomodar o bocal do aparelho telefônico.

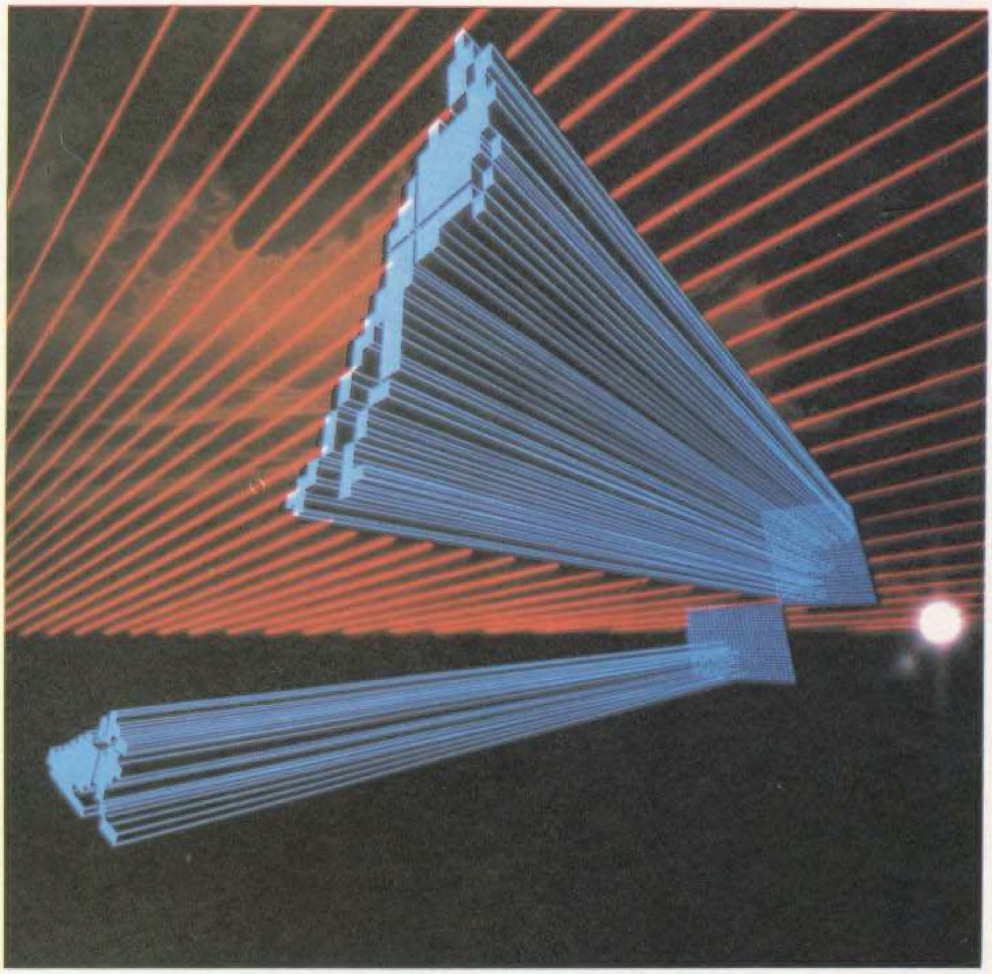
O acoplador acústico é conectado ao micro e transforma seus sinais em tons que podem ser enviados pela linha telefônica. Ele também converte tons de entrada vindos de outro computador em informações digitais que o computador receptor pode entender.

Já o modem de conexão direta codifica (ou modula) o dado do computador diretamente em sinais elétricos e decodifica (ou demodula) a informação de entrada em bits seriais entendidos pelo computador. Esses modems podem transmitir informações a velocidades maiores que os acopladores acústicos e são menos propensos a erros.

COMO FUNCIONA O GERADOR GRÁFICO

- AS POSIÇÕES DE IMPRESSÃO
- MOVIMENTO BLOCO POR BLOCO
- COMO ABRIR CANAIS
- USE TABELAS DE REFERÊNCIA
- COMO MODIFICAR APONTADORES

Incluído nos programas de animação gráfica para os micros das linhas Spectrum e TRS-Color que apresentamos aqui, o gerador gráfico descomplicará a sua vida.



Em artigo anterior da seção *Código de Máquina*, instruímos os usuários do TK-90X e do TRS-Color a inserirem dados para a montagem de um quadriculado que poderia ser usado na criação de caracteres gráficos (UDG). Nesse artigo, apresentamos um pequeno programa em linguagem de máquina no qual foram inseridos alguns números cujo significado — afirmávamos — não precisava ser compreendido naquele momento.

Mas agora, com os conhecimentos que você já adquiriu a respeito da linguagem de máquina, é possível abordar o que significam esses números. Para facilitar sua compreensão, decodificamos os números que estavam em Assembler, transformando-os em mnemônicos.

Os usuários do MSX, do Apple e do TK-2000 não precisam se preocupar com isso, pois os programas para esses micros empregam sprites e figuras móveis para criar os caracteres gráficos, não exigindo assim rotinas em código de máquina.

```
 org 65200
     jr start
mode defb 1
data defb 22
     defb 0
     defb 0
     defb 32
     defb 32
     defb 32
     defb 22
     defb 0
     defb 0
     defb 32
     defb 32
     defb 32
     defb 22
     defb 0
     defb 0
     defb 32
     defb 32
     defb 32
     defb 22
     defb 0
     defb 0
     defb 144
     defb 145
     defb 146
     defb 22
     defb 0
     defb 0
     defb 147
     defb 148
     defb 149
     defb 22
     defb 0
     defb 0
     defb 150
     defb 151
     defb 152
     defb 22
     defb 0
     defb 0
     defb 153
     defb 154
     defb 155
     defb 22
     defb 0
     defb 0
     defb 156
     defb 157
     defb 158
     defb 22
     defb 0
     defb 0
     defb 159
     defb 160
     defb 161
start ld a,(mode)
     cp 1
     ld bc,18
     jr z,print
     jr c,reset
     sla c
     jr print
reset ld c,0
print ld ix,data
     add ix,bc
```

```
ld a,(23689)
ld b,a
ld a,24
sub b
ld (ix+1),a
inc a
ld (ix+7),a
inc a
ld (ix+13),a
ld a,(23688)
ld b,a
ld a,33
sub b
ld (ix+2),a
ld (ix+8),a
ld (ix+14),a
push ix
ld a,2
call $1601
pop de
ld bc,18
call $203C
ret
```

A primeira instrução faz com que o processador passe para o início do programa propriamente dito, saltando sobre a longa lista de dados que a segue. Evidentemente, poderíamos chamar o programa na primeira instrução que aparece após a lista de dados; mas, como veremos a seguir, é mais fácil chamá-lo pelo endereço de origem.

O primeiro byte de dados, situado logo depois do rótulo **mode**, diz ao programa que quadro deve ser usado. O BASIC dá um **POKE** 0, 1 ou 2 nessa posição. O UDG 0 é um quadro vazio que serve para limpar a tela; e os quadros 1 e 2 são caracteres UDG (por exemplo, tanques de guerra, cada um apontando para uma direção). Se nenhum número for especificado, a rotina assumirá o valor 1 para o quadro.

OS DADOS DO UDG

Depois do rótulo **DATA**, aparecem os detalhes de três quadriculados. O Spectrum desenha seus UDG na forma de cadeias, mas primeiro tem que ser informado sobre em que lugar colocá-los. As instruções para isso também estão contidas na cadeia de dados: 22 é o código para o **AT** em BASIC e os dois zeros são bytes vazios que a rotina utilizará mais tarde para definir a posição de impressão.

As três séries de três números 32 formam o quadro vazio de 9 x 9. O número 32 é o código ASCII para o caractere de espaço. Cada série de três forma uma linha do quadro e deve ser precedida pelo seu próprio **AT** e pela posição de impressão. Esse é o quadro 0.

O quadro 1 é feito dos UDG de 144 a 152. Nos dados, eles vêm em grupos de três com um **AT** e dois bytes livres para a posição de **print**. O quadro 2 é feito dos UDG de 153 a 161.

QUAL UDG?

O **ld a, (mode)** carrega o número inserido, via **POKE**, no byte **mode** do acumulador, e **cp 1** o compara com 1.

O par de registradores BC é então carregado com 18. Na realidade, o número 18 vai para o registrador C, enquanto o B fica vazio. O que interessa durante a rotina principal é o valor do registrador C; mas o registrador vazio B será usado mais tarde, quando chamarmos rotinas da ROM.

Se o número do **mode** carregado no acumulador for 1, **cp 1** ajustará para esse número o indicador de zero. Então, **jr z, print** saltará para a rotina de nome **print**. Lá, o registrador IX é carregado com o endereço do primeiro byte de dados do UDG e somado com o conteúdo do par BC, 18.

Cada quadro contém dezoito bytes de dados: nove para caracteres UDG, três para os **AT** e seis para posições de impressão. Isso faz o registrador IX passar sobre o quadro 0 e ir para o começo do quadro 1. Se o número do **mode** for 0, **cp 1** ligará o indicador de carry. Em seguida, **jr c, reset** saltará para o rótulo **reset** e carregará C com 0. A instrução **add ix, bc** somará então IX ao endereço do rótulo **data**, deixando IX apontando para o começo do quadro vazio.

Se o número do **mode** não for 0 nem 1, deverá ser 2. Nesse caso, o processador vai para a instrução **sla c**. Esta significa "rotação aritmética para a esquerda" e age sobre o registrador C. Ela move todos os bits uma casa à esquerda, multiplicando por 2 o conteúdo do registrador.

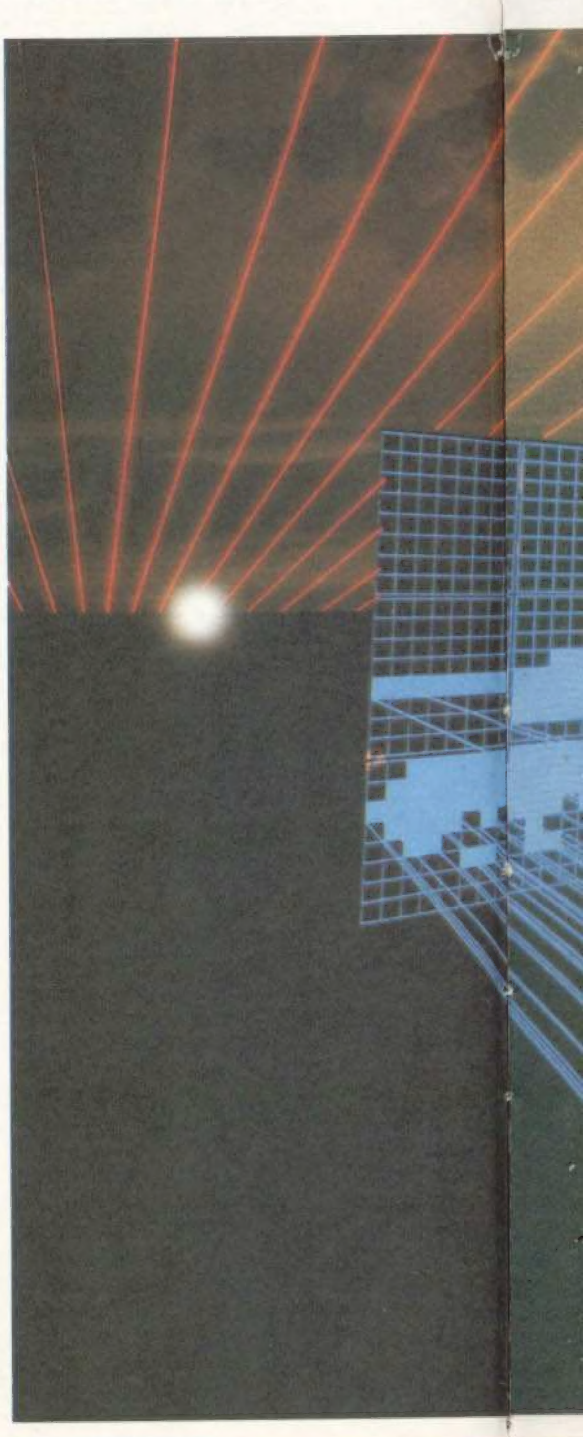
Assim, ela duplica o 18 (que vai para 36) e salta para o rótulo **print**. Quando BC for somado ao apontador IX, este último será transferido 36 bytes acima na lista de dados, parando na posição do começo do quadro 2.

AS POSIÇÕES DE IMPRESSÃO

A variável de sistema em 23 689 contém a posição de impressão vertical. Ora, a posição de impressão especificada pelo BASIC é a do topo do quadro. Essa variável conta de 1 a 24 de baixo para cima da tela, e não ao contrário, como em BASIC. Portanto, ela deve ser transferida para o registrador B; depois, 24 é carregado em A e o valor de B é subtraído dele.

O apontador IX contém o endereço do primeiro byte do quadro que a rotina irá desenhar. Então **ld (ix + 1), a** carrega a posição vertical de impressão do topo do quadro no próximo byte da lista de dados (o primeiro 0 que estiver vazio).

A é então incrementado para fornecer a posição vertical de impressão da próxima linha de caracteres UDG do quadro. Isso é carregado no oitavo byte da lista por **ld (ix + 7), a**. Depois, A é incrementado novamente para a terceira



linha e isso é carregado no décimo quarto byte.

Essa operação carregou as posições verticais de impressão nos bytes vazios logo após o código do **AT**, 22.

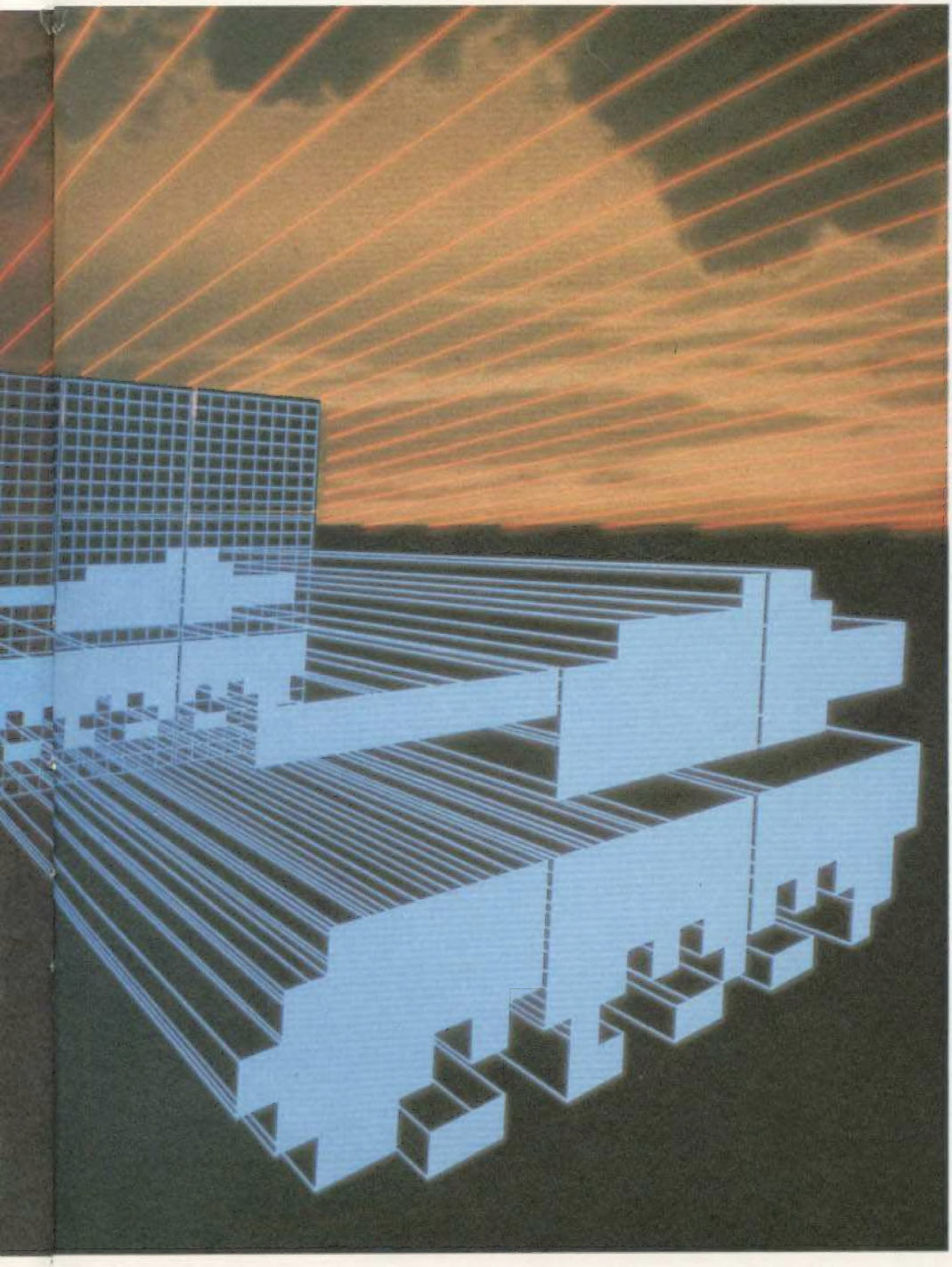
As posições horizontais de impressão provenientes da variável de sistema 23 688 são então carregadas no acumulador. Mais uma vez, essa operação é feita de trás para a frente, se comparada ao que acontece no BASIC. Portanto, a posição horizontal é carregada em B; A é carregado com 33 e B é subtraído de A para fornecer corretamente a posição de impressão.

Como cada linha começa na mesma posição horizontal, isso significa que o mesmo valor de A pode ser carregado em outros espaços vazios na lista de dados. As instruções adequadas para fazer isso são **ld (ix+2),a**, **ld (ix+8),a** e **ld (ix+14),a**.

O apontador IX é então "empurrado" para a pilha, protegido contra uma eventual alteração causada pela rotina da memória ROM que está prestes a ser chamada. A é carregado com 2, e a rotina de abertura do canal em 1 601 é chamada. O 2 no acumulador define o canal a ser usado quando o processador vai para uma sub-rotina. O canal 1 é a linha de edição na parte inferior da tela; o canal 2 é a tela principal; é o 3 a impressora.

O apontador IX que foi guardado na pilha é então recuperado no par DE. Depois, BC é carregado com 18, e a rotina de impressão de cadeia em 2 032 é chamada. Essa rotina imprime BC caracteres, começando em DE. Portanto, ela imprime os dezoitos caracteres que compõem o quadro, a começar do endereço base que está no apontador IX.

Feito isto, **ret** retorna o processador para o BASIC.



```
        ORG 32000
        LDX 32700
        LDA #3
        STA FCNT
        STA SCNT
        LDA #8
        STA TCNT
        LDA 32250
        BEQ JUMP
        LDU #32300
        DECA
        LDB #72
        MUL
        LEAU D,U
LOOP    LDA ,U+
        STA ,X
        LEAX 32,X
        DEC TCNT
        BNE LOOP
        LDA #8
        STA TCNT
        LEAX -255,X
        DEC FCNT
        BNE LOOP
        LDA #3
        STA FCNT
        LEAX 253,X
        DEC SCNT
        BNE LOOP
        RTS
JUMP    CLRB
STLP    STB ,X
        LEAX 32,X
        DEC TCNT
        BNE STLP
        LDA #8
        STA TCNT
        LEAX -255,X
        DEC FCNT
        BNE STLP
        LDA #3
        STA FCNT
        LEAX 253,X
        DEC SCNT
        BNE STLP
        RTS
FCNT    RMB 1
SCNT    RMB 1
TCNT    RMB 1
```

O endereço de tela correspondente ao canto superior esquerdo do quadriculado é armazenado nas posições de memória 32 700 e 32 701 pelo programa BASIC; **LDX** carrega esse apontador de dois bytes no registrador de X dezesseis bits. O número 3 é carregado no acumulador e armazenado nos dois contadores de nomes **FCNT** e **SCNT**. O quadriculado contém 3 x 3 caracteres.

Esses contadores aparecem na lista de dados no fim do programa. Aqui, o espaço que o contador vai ocupar é reservado por **RMB** (reserva byte de memória). O número 1 que vem a seguir significa que somente um byte de memória deve ser reservado. Assim, **RMB 50** reservaria 50 bytes de memória.

O terceiro contador, **TCNT**, é ajustado para 8. Depois, o conteúdo de 32 250 é carregado em A. O número do UDG é armazenado em 32 250; se esse número for zero, **BEQ JUMP** desviará o programa para uma rotina que limpa a tela naquela área. Caso contrário, o processador continuará trabalhando com o UDG propriamente dito.

IMPRIMA BLOCOS GRÁFICOS

O lugar da memória onde os UDG ficam guardados começa na posição 32 300; esse número é armazenado em U para facilitar os cálculos que definem onde se inicia o UDG. O primeiro caractere começa no princípio dessa área; portanto, seu equivalente é 0. Note que, no acumulador, o número de caractere pedido é decrementado.

LDB #72 carrega o registrador B com 72, e **MUL**tiplica os conteúdos de A e B, guardando o resultado em D. Os acumuladores A e B são registradores de oito bits; D é um registrador de dezesseis bits; por isso, ele não é sobrecarregado com o resultado. Existem 72 bytes em cada gráfico. Cada caractere contém oito bytes e há nove caracteres para cada quadriculado (9 x 8 = 72). Essa operação calcula o fator necessário para se achar o endereço inicial do UDG apropriado.

A instrução **LEAU D,U** carrega U com o valor de D mais U. Em outras palavras, ela vai contando ao longo da lista até que seja apontado o começo do caractere pedido.

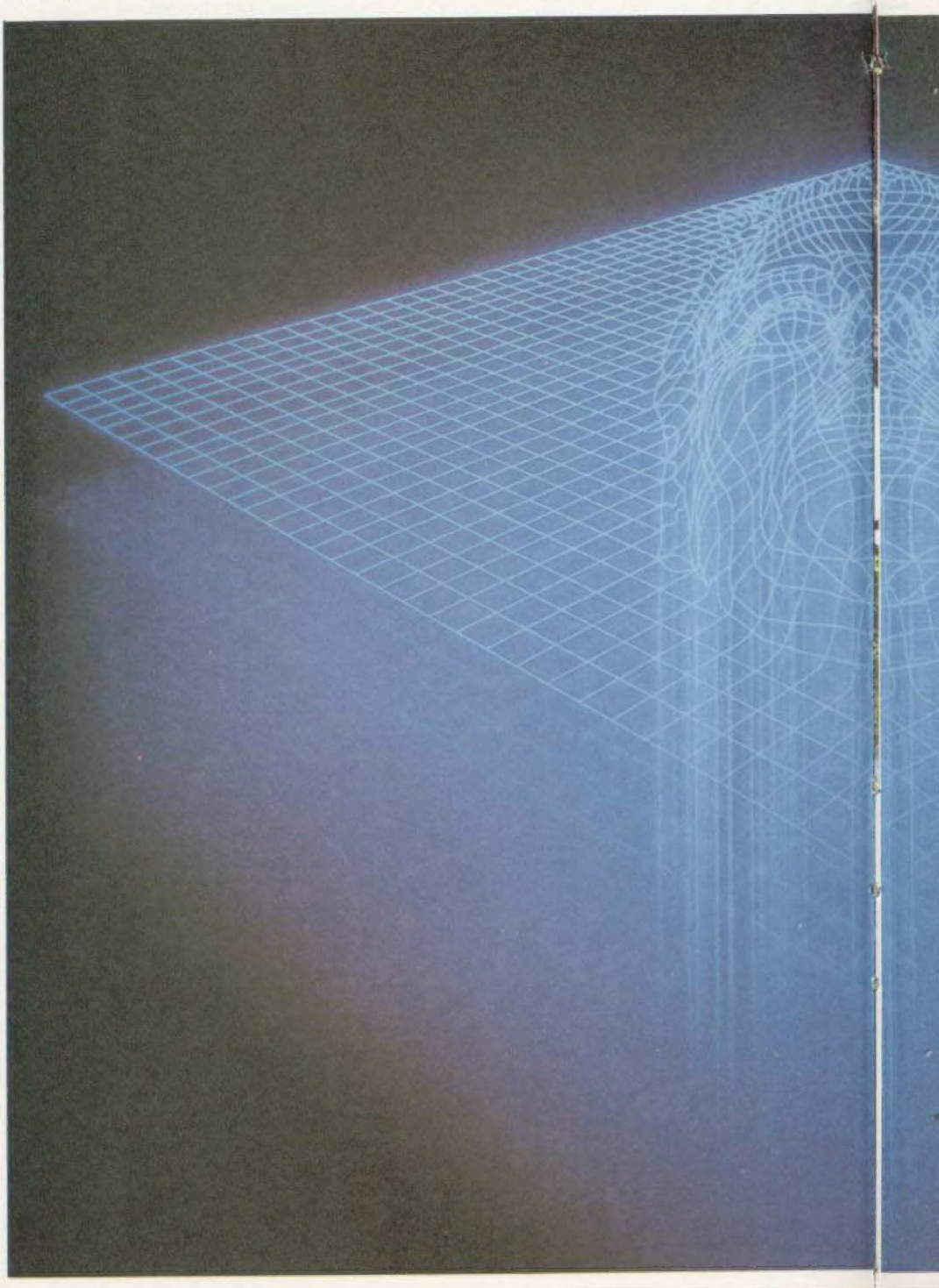
LDA ,U + carrega o acumulador com o conteúdo daquela posição de memória; depois, incrementa o registrador U e prepara-se para trabalhar com o próximo. **STA ,X** armazena o byte do caractere obtido da lista de gráficos na posição de memória apontada por X. Lembremos que o registrador X contém a posição do canto superior esquerdo do bloco na tela gráfica.

LEAX 32,X soma 32 ao registrador X; isso move o apontador para a próxima linha (há 32 posições em cada linha). O contador **TCNT** é então decrementado; se ele não for igual a zero, a instrução **BNE** retornará ao endereço **loop** para trabalhar com o próximo byte do caractere.

O valor inicial de **TCNT** é 8; por isso, **loop** deve ser executado oito vezes. Precisamos de oito bytes, dispostos uns sobre os outros, para formar um caractere. Podemos chamar **TCNT** de contador de bytes de um caractere.

ADICIONE OUTRO CARACTERE

LDA #8 e **STA TCNT** retornam o valor 8 para o contador **TCNT**, que já está pronto para trabalhar com o próximo caractere.



Em seguida, a instrução **LEAX -255,X** move o apontador X para o início do caractere à direita.

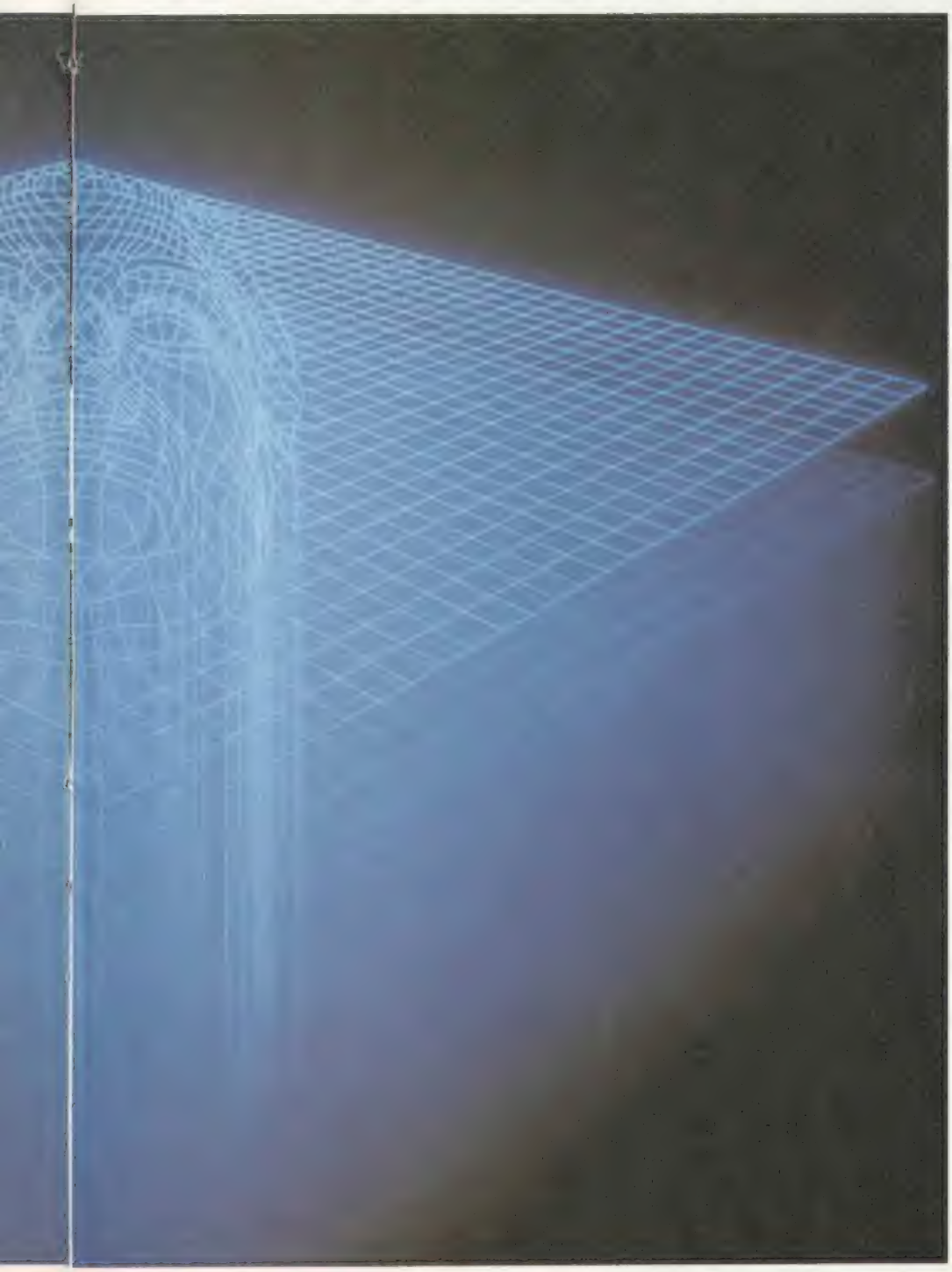
Devemos lembrar ainda que o registrador X contém o apontador de tela, e foi oito vezes incrementado de 32 para imprimir totalmente o caractere (32 x 8 = 256).

O contador **FTNT** é decrementado de modo a percorrer toda a matriz formada por nove caracteres (ou seja, três de cada lado). Quando o resultado do decrementado for diferente de 0 (zero), o processador voltará para **loop** e trabalhará com o primeiro byte do caractere seguinte.

Se o resultado for 0 (zero), o processador continuará pelo programa.

COMO PERCORRER O QUADRICULADO

O contador **FTNT** é então reajustado para 3 e o apontador X, adiantado 253 posições. Lembre que, anteriormente, ele havia sido adiantado 32 posições, o que fez com que o apontador fosse para a posição de tela diretamente abaixo do caractere que acabara de ser impresso. Em seguida, ele foi atrasado em 255, o que o deixou na posição de tela imediatamente à direita do topo do último caractere.



Se desenharmos um pequeno quadriculado mostrando os caracteres e observarmos as posições relativas das locações de tela, veremos que o registrador X foi adiantado em 32 posições. Ele aponta agora a posição de tela diretamente abaixo do último caractere impresso e está duas locações à direita do lugar em que deveria estar o primeiro byte do primeiro caractere na segunda fileira.

Portanto, devemos subtrair 2 do apontador X, a fim de movê-lo para a posição correta.

Mas, como o programa já havia subtraído 255, basta somar 253 que iremos para o lugar certo.

O contador **SCNT** é decrementado; ele conta as três linhas de caracteres que formam o gráfico inteiro. Quando o resultado do decremento for diferente de 0 (zero), o **BNE** levará o processador de volta para trabalhar com a linha seguinte.

Todavia, se o resultado for 0 (zero), isso significa que o gráfico foi impresso totalmente; nesse caso, **RTS** levará o processador de volta para o programa BASIC.

Siga atentamente as instruções, que o processo acabará se tornando claro e compreensível.

LIMPE A TELA

Quando o número do bloco a ser impresso na tela for igual a zero, a rotina simplesmente limpará a área da tela que está sendo acessada. O modo pelo qual se limpa a tela é praticamente o mesmo pelo qual se imprime nela. A diferença entre eles é que a rotina não precisa consultar a lista de caracteres, mas somente imprimir o código 0, ou seja, nada.

CLRB ajusta o registrador B para 0. O 0 é então armazenado nas posições de tela apontadas pelo registrador X. Se você quiser se movimentar pelo quadriculado, deve atualizar repetidamente o registrador X, da mesma maneira que na rotina de impressão; só que, desta vez, o 0 em B é armazenado em cada posição. Uma vez mais, o **RTS** retornará o processador ao BASIC para rodar o resto do programa.

CONJUNTOS DE BLOCOS GRÁFICOS (3)

Não se trata de mais uma fábula de Esopo ou de La Fontaine: aqui, o macaco e a cobra se reúnem na floresta para nos ensinar os segredos do trabalho com blocos gráficos.

Nos artigos anteriores, aprendemos a criar um grande número de blocos gráficos e começamos a trabalhar com grupos de caracteres para fazer um cenário de floresta.

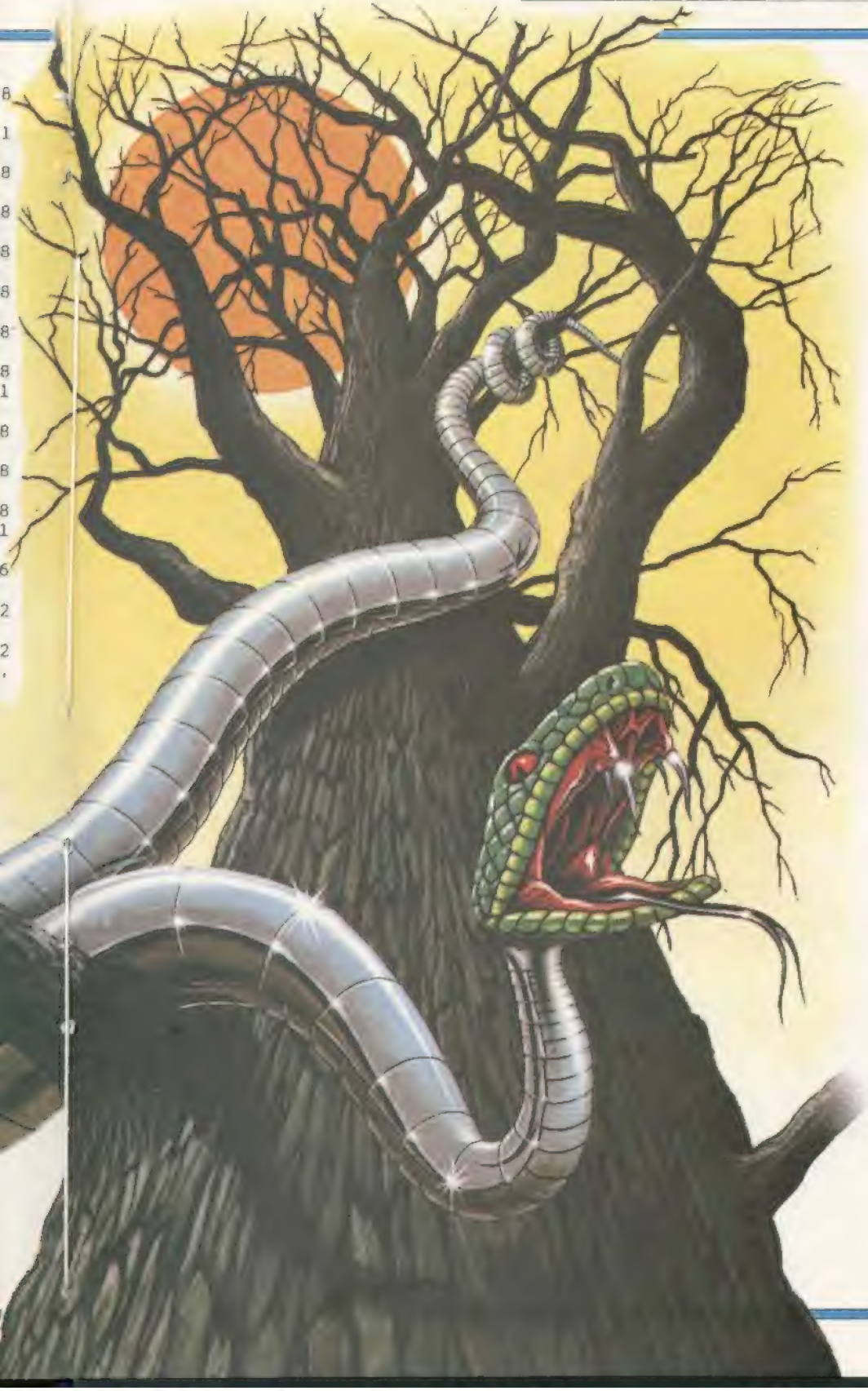
O programa que faz o desenho no último artigo está dividido em partes (com exceção do Apple e do TK-2000, para os quais publicamos apenas o banco de blocos a ser digitado por intermédio do monitor). Cada seção cuida de um segmento (ou figura) do desenho — crocodilo, elefante, árvores, e assim por diante. A seção deste artigo acrescenta uma cobra e um macaco, além de concluir o fundo do quadro. Tanto no caso do Apple como no do TK-2000, apresentamos o programa completo.

Carregue agora a primeira parte do programa para poder digitar a continuação. Os usuários do Apple e do TK-2000 devem antes carregar o banco de blocos gravado no artigo anterior por intermédio do monitor.

```
140 POKE 23676,254
150 FOR n=USR "a" TO USR "r"+7
: READ a: POKE n,a: NEXT n
160 POKE 23676,253
170 FOR n=USR "a" TO USR "j"+7
: READ a: POKE n,a: NEXT n
190 POKE 23676,252
200 FOR n=USR "a" TO USR "u"+7
: READ a: POKE n,a: NEXT n
210 POKE 23676,251
220 FOR n=USR "a" TO USR "u"+7
: READ a: POKE n,a: NEXT n
230 POKE 23676,250
240 FOR n=USR "a" TO USR "g"+7
: READ a: POKE n,a: NEXT n
510 INK 1
520 POKE 23676,254
530 LET z=144: FOR n=2 TO 7:
FOR m=16 TO 18: PRINT AT m,n;
CHR$ z: LET z=z+1: NEXT m:
NEXT n
540 POKE 23676,253
550 LET z=144: FOR n=8 TO 12:
FOR m=17 TO 18: PRINT AT m,n;
CHR$ z: LET z=z+1: NEXT m:
NEXT n
610 INK 0
620 POKE 23676,252
630 PRINT AT 0,30;CHR$ 144;AT
1,24;CHR$ 145;CHR$ 146;AT 1,28
;CHR$ 147;CHR$ 148
640 PRINT AT 2,23;CHR$ 149;
CHR$ 150;CHR$ 151;CHR$ 32;CHR$
152;CHR$ 153
650 PRINT AT 3,23;CHR$ 154;
CHR$ 155;CHR$ 32;CHR$ 32;CHR$
156;CHR$ 157;CHR$ 158
660 PRINT AT 4,24;CHR$ 159;
CHR$ 160;CHR$ 32;CHR$ 161;CHR$
162;CHR$ 163;CHR$ 164
670 POKE 23676,251
680 PRINT AT 5,24;: FOR n=144
TO 151: PRINT CHR$ n;: NEXT n
690 PRINT AT 6,24;CHR$ 152;
CHR$ 153;CHR$ 153;CHR$ 153;
CHR$ 153;CHR$ 154;CHR$ 155;
CHR$ 156
700 PRINT AT 7,23;CHR$ 157;
CHR$ 158;CHR$ 159;CHR$ 32;CHR$
32;CHR$ 160;CHR$ 161;CHR$ 162;
AT 8,22;CHR$ 163;CHR$ 164;
710 POKE 23676,250
720 PRINT CHR$ 144;CHR$ 145;
CHR$ 32;CHR$ 32;CHR$ 146;CHR$
147;AT 9,24;CHR$ 148;CHR$ 149;
AT 10,24;CHR$ 150
855 INK 6
860 FOR n=0 TO 2*PI STEP .05:
PLOT 70,150: DRAW SIN n*12,
COS n*12: NEXT n
870 FOR n=0 TO 2*PI STEP PI/4:
PLOT 70,150: DRAW COS n*20,SIN
n*20: NEXT n
970 INK 0
```

```
10 FOR I=0 TO 983
150 CIRCLE (50,30),25,11:PAINT
(50,30),11
160 DRAW "BM50,30C11NU30NE60NR8
4NF60ND84NG60NL84NH30"
200 FOR I=1 TO 131
3090 DATA 641,18,210,673,19,210
,705,20,210,642,21,210
3100 DATA 674,22,210,706,23,210
,643,24,210,675,25,210
3110 DATA 707,26,210,644,27,210
,676,28,210,708,29,210
3115 REM APAGAR
3120 DATA 645,30,210,677,31,210
,709,32,210,646,33,210
3130 DATA 678,34,210,710,35,210
,679,36,210
3135 REM APAGAR
3140 DATA 711,37,210,680,38,210
,712,39,210,681,40,210
3150 DATA 713,41,210,682,42,210
,714,43,210,683,44,210,715,45,2
10
3160 DATA 23,46,148,49,47,148,5
0,48,148,53,49,148
3170 DATA 54,50,148,80,51,148,8
1,52,148,82,53,148
3180 DATA 84,54,148,85,55,148,1
12,56,148,113,57,148
3190 DATA 116,58,148,117,59,148
,118,60,148,145,61,148
3200 DATA 146,62,148,148,63,148
,149,64,148,150,65,148
3210 DATA 151,66,148,177,67,148
,178,68,148,179,69,148
3220 DATA 180,70,148,181,71,148
,182,72,148,183,73,148
3230 DATA 184,74,148,209,75,148
,210,76,148,211,76,148
3240 DATA 212,76,148,213,76,148
,214,77,148,215,78,148,216,79,1
48
3250 DATA 240,80,148,241,81,148
,242,82,148,245,83,148
3260 DATA 246,84,148,247,85,148
,271,86,148,272,87,148
3270 DATA 273,88,148,274,89,148
,277,90,148,278,91,148,305,92,1
48,306,93,148,337,94,148
3280 DATA 500,95,226,501,96,226
,502,97,226,503,98,226
3290 DATA 532,99,226,533,100,22
6,534,101,226,535,102,226
3300 DATA 536,103,226,565,104,2
26,566,105,226,567,106,226,568,
107,226
```

- LINHAS **DATA** QUE COMPLETAM O CENÁRIO DA FLORESTA
- UM MACACO E UMA COBRA
- O PROGRAMA COMPLETO NO APPLE E NO TK-2000
- COMO COMPLETAR O FUNDO
- SUGESTÕES PARA ANIMAÇÃO
- COMO GUARDAR O BANCO DE BLOCOS
- O SOL EM ALTA RESOLUCÃO



```
3310 DATA -32,-4,36,75,80,147
3320 DATA 342,108,36,343,109,36
,374,110,36,375,111,36
3330 DATA 406,112,36,407,113,36
,439,114,96,471,114,96
3340 DATA -44,-20,-15,60,84,96,
157,163
3350 DATA 327,115,36,328,116,36
,329,117,36,330,118,36
3360 DATA 360,119,96,361,120,96
,392,121,96,424,121,96
3370 DATA 456,122,96
```

```
1400 REM COBRA
1410 DATA 0,0,0,0,0,0,3,14,31,3
1,63,67,65,254,254,225,71,63,0,
0,0,0,0,0,0
1420 DATA 0,0,0,0,255,12,28,28,
28,254,143,1,0,4,248,208,224
1430 DATA 0,0,0,0,0,0,0,0,0,0,0
,128,240,56
1440 DATA 52,99,225,225,225,96,
47,31,14,6,2,1,0,0,0,0,0,0,0,0,
3,5,63,127
1450 DATA 255,225,240,248,252,2
55,255,126,60,28,124,247,247,11
5,31,1,0,0,0,0
1460 DATA 255,191,159,143,199,1
94,127,64,128,128,64,63
1470 DATA 48,112,248,249,253,25
5,189,255,0,0,0,0,192,252,190,6
3,63,24,240
1480 DATA 9,5,7,5,249,67,243,25
1,255,252,238,239,255
1490 DATA 192,224,208,223,224,1
29,193,227,243,247,255,248,240,
224,192
1500 DATA 0,0,0,0,255,129,195,1
99,199
1510 DATA 239,239,255,0,0,0,0,0
,0,0,0,128,248,31,140,158,191,1
91,255,0,0,0,0
1520 DATA 0,0,0,1,15,62,206,30,
61,121,243,247,247,63,15,3,0
1530 DATA 30,127,191,95,93,61,2
50,240,194,252,194,192,192,224,
224,240
1540 REM MACACO
1550 DATA 1,2,4,8,16,32,64,128,
0,0,0,3,31,63,127,252,0,0,0,224
,248,252,252
1560 DATA 62,3,15,15,15,15,14,1
4,12
1570 DATA 3,142,240,192,128,0,0
,0,1,1,3,3,3,3,3,3,248,240,224,
224,192
1580 DATA 192,192,192,30,14,110
,76,56,0,0,0,0,0,0,0,0,0,0
```

```
1590 DATA 1,28,28,56,56,112,112
,224,224
1600 DATA 3,3,1,1,1,0,0,0,192,2
24,224,240,248,248,252,126,3,7,
15,31,63,62
1610 DATA 126,252,192,192,128,0
,0,0,3,7,0,0,0,0,0,0,128,223
1620 DATA 127,63,31,31,15,7,7,3
,0,128,128,192,192,192,224,240,
252,126,126,127
1630 DATA 63,63,31,31,15,15,7,3
,3,131,131,131
1640 DATA 255,255,231,231,255,2
52,172,183,128,192,224,224,224,
224,224,224,224
1650 DATA 1,1,0,0,0,1,3,7,248,2
55,255,255,255,255,255,255,0
1660 DATA 255,255,255,255,255,2
55,255
1670 DATA 15,255,255,255,255,25
5,255,255,199,255,255,255,255,2
55,255,255
1680 DATA 223,239,255,255,252,1
28,0,128,224,240,248,248,240,96
,0,0,0,0,200
1690 DATA 232,248,248,120,112
1700 DATA 7,15,15,31,31,31,63,6
3,255,255,255,255,255,255,255,2
55,192,224,224
1710 DATA 224,224,224,192,192,0
,0,1,1,3,7,14,60,240,224,192,12
8,128,0,0,0
1720 DATA 0,0,0,0,0,3,15,31,63,
63,63,63,126,254,252,240,191,19
1,63,63,63
1730 DATA 63,63,63,15,7,7,7,7,7
,7,7,192,131,135,159,191,255,25
4,252
1740 DATA 252,248,240,224,192,1
28,0,0,0,1,3,7,6,6,4,0,127,254,
240,224
1750 DATA 64,64,0,0,192,0,0,0,0
,1,1,3,63,31,63,127,254,252,248
,240
1760 DATA 7,3,0,0,0,0,0,0,240,1
92,0,0,0,0,0,0,7,7,15,31,31,31,
23,23,224
1770 DATA 192,128,0,0,0,0,0,7,6
,4,0,0,0,0,0
```

```
20 DIM C(59),M(156),S(48),E(17)
,T1(1),T2(7),F1(7),F2(7)
90 PCLS 3
100 DRAW "BM62,1C2DG6LG2LGLUHLG
LD6GDGDGDG7DG2DRD3RD2FDFDL12H2L
ULU3HUH4UH2ULU5EUE4R3F3D2GLHURB
D2R2E2U3HUHLHL4GL2G4DGD7FD2FDFD
F2D"
110 DRAW "F2DFD2G3DGDGD2GD5G2L2
GLG2LGLG2D3EUER2ND2R2R2ER2ERERE
UEUEUERD9FG4DG2DG2D2ND2R2D5E2U5
E8U9E3R17F3D6FR3EREREE2RE5UE3UE3
UE"
120 DRAW "U2NU2LH2LD3FDGDGDG4LG
LGLG2LG2HU2EUEU4H3E2R4ERFRFEURU
H2U6H3L4GH2L2G2DF2D4G2L2H2U2HUH
UHUEUEUE2UEUEUEUEU2EUEREUL4D5L3
D6L2D3L2D3"
130 DRAW "BM24,16U3NL6DL8D2L2D2
L2D9BM70,43D4G8BM3,64E3BM+10,10
D3"
140 PAINT (50,50),2
150 DRAW "BM56,34C1RDBF3DBL7DF2
BE2BU2"
160 GET(0,0)-(76,80),M,G
170 PCLS
180 DRAW "BM13,11C4G3L8H2U2EUE2
ERERER8FR2F3RE3RERER11FR3F2RF4R
12FR3FR4ERER2ERERER3FD4GBU3GDG2
LGFNR3GD2FDFL4"
190 DRAW "H2LHLHLHL18G4LGL16HL3
HLH9LHL2HL2G3DFR3BM23,6D3FDF5RF
2R9ER2ER7BM52,8LGL3G2L11H2UE2R7
F4"
200 PAINT (60,10),3,4:PAINT (13
,5),3,4
210 DRAW "BM83,8C1DBL4BG5H3G4H5
G5H4G5H3G4H4G3H8E4F2E2F2E2F2E2F
4BM7,13LH3E6F4E4FDFD2F9R2FR15E"
220 GET(0,0)-(88,21),S,G
370 CIRCLE(50,30),25,2:PAINT (5
0,30),2:DRAW "BM50,30C2NU30NE60
NR84NF60ND84NG60NL84NH30"
430 PUT (10,166)-(98,187),S,PSE
T
440 PUT (140,0)-(216,80),M,PSET
```

Essas adições encaixam direitinho nos programas do artigo anterior — exceto no MSX, em que várias linhas devem ser substituídas e duas delas apagadas. O que fizemos foi criar mais blocos gráficos. Novamente as linhas **DATA** de 1 400 a 1 770 são comuns ao Spectrum e ao MSX. O programa do Spectrum começa com novas linhas que colocarão na memória os dados adicionais, usando o comando **POKE**.

A listagem do MSX teve o comprimento do laço da linha 10 mudado, a fim de que os novos dados fossem colocados na parte alta da memória. As linhas 150 e 160 desenham o sol. O laço

da linha 200 também teve seu comprimento modificado para que as posições e cores dos novos blocos pudessem ser lidas. Essas novas posições e cores se encontram nas linhas **DATA** a partir de 3 090. Observe que parte das linhas do antigo programa teve que ser apagada, pois as posições dos blocos no banco foram modificadas.

O TRS-Color usa **DRAW** para desenhar os novos personagens e **GET** para armazenar os padrões em matrizes.

Cada programa coloca os blocos nas posições adequadas no cenário.

O primeiro passo consiste em carregar o banco de blocos fornecido, por meio do monitor. Se você gravou o seu, entre o monitor com:

```
CALL -151
```

posicione a fita e digite:

```
4000.5000 L
```

Ou recorra ao gerador de blocos, se o seu Apple usa disquete. Eis o programa completo:

```
10  HGR :E = 16384:T = 8192
20  HCOLOR= 1: FOR I = 0 TO 176
: HPLOT I,159 - 4 *  SQR (176 -
 I): NEXT
30  FOR I = 120 TO 279: HPLOT I
,130 - 6 *  SQR (I - 120) + (I
- 200) * (I > 200) / 10: NEXT
40  HCOLOR= 5: FOR I = 0 TO 7 S
TEP .05:F = ( INT ( RND (1) * 1
0) = 5)
50  HPLOT 50 + (30 + 20 * F) *
 COS (I),50 + (26 + 20 * F) *
SIN (I) TO 50 + (30 + 20 * F) *
  COS (I + 3.14),50 + (26 + 20
* F) *  SIN (I + 3.14): NEXT I
60  READ NF: FOR M = 1 TO NF: R
EAD L,NL
70  FOR J = 1 TO NL: READ BI,C,
NB: FOR K = 1 TO NB
80 Y = L + J - 1:X = C + K - 1:
N = BI + K - 1
90  FOR I = 0 TO 7
100  POKE T + (Y - 8 * (Y > 7)
- 8 * (Y > 15)) * 128 + 40 * (Y
 > 7) + 40 * (Y > 15) + X + 102
4 * I, PEEK (E + N * 8 + I)
110  NEXT I,K,J,M
120  END
150  POKE T + (Y - 8 * (Y > 7)
- 8 * (Y > 15)) * 128 + 40 * (Y
 > 7) + 40 * (Y > 15) + X + 102
4 * I, PEEK (E + N * 8 + I)
1000  DATA      7,17,3,1,25,14,4
1,25,14,81,25,14
1010  DATA  17,3,160,4,7,200,4,
13,240,4,13
1040  DATA      3,7,30 ,30,9,70,
30,9,110,30,9,154,34,1,194,34,1
,234,34,1,273,33,3
1050  DATA      5,7,30,32,7,70,3
2,7,110,32,7,154,36,1,194,36,1,
234,36,1,273,35,3
1060  DATA      5,7,30,28,9,70,2
8,9,110,28,9,154,32,1,194,32,1,
234,32,1,273,31,3
1070  DATA      7,7,30,30,9,70,
30,9,110,30,9,154,34,1,194,34,1
,234,34,1,273,33,3
1080  DATA      0,10,25,25,1,5
8,18,7,97,17,7,137,17,7,178,18,
8
1090  DATA      219,19,9,258,
18,9,297,17,9,15,17,8,57,19,1
```

No TK-2000 entramos no monitor por meio de **LM** (e o banco de blocos deve ser carregado em outra posição):

```
A000.B0000 L "nome"
```

A floresta no TRS-Color.

Na linha 10, o valor de T deve ser mudado para 40 960. Os usuários desse micro também podem mudar as cores nas linhas 20 a 40. Para modificar as cores dos animais é preciso alterar de uma unidade a coluna na qual é impressa a figura. (Azul e verde sairão invertidos no TK-2000.)

COMO FUNCIONA

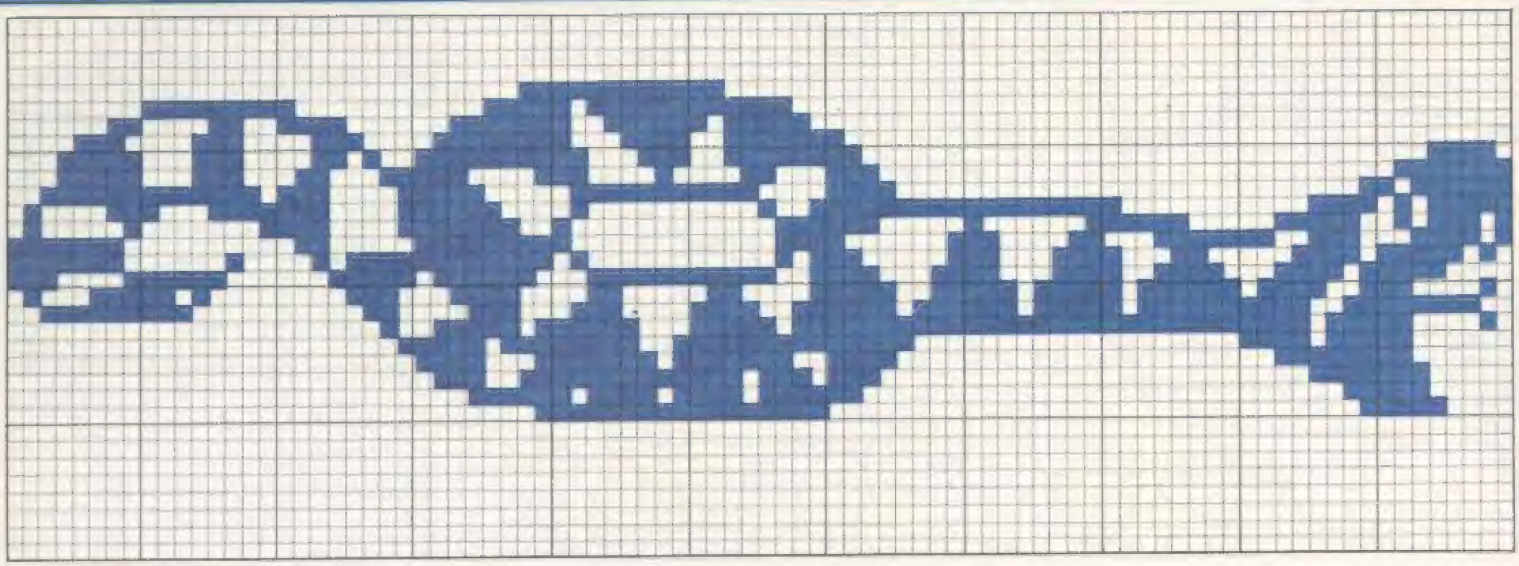
A linha 10 liga a tela gráfica e armazena em E e T os endereços iniciais das duas páginas de vídeo.

As linhas 20 e 30 desenham as colinas, e as linhas 40 e 50, o sol. Como não há comandos específicos para traçar elipses no Apple, tivemos que lançar mão de fórmulas matemáticas.

As linhas 60 a 120 usam quatro laços **FOR ... NEXT** para colocar os blocos em posição correta na tela. Os dados necessários para tanto estão nas linhas **DATA** de 1 000 em diante.

O método posto em prática é o seguinte: inicialmente, a linha 60 usa **READ** para descobrir o número de figuras (**NF**) que serão desenhadas. O laço **FOR ... NEXT** que se inicia nessa linha cuida para que o processo de desenhar uma figura seja repetido **NF** vezes. A cada figura desenhada, são obtidos a seguir, também com **READ**, a linha (**L**, entre 0 e 39) em que vai ser colocado o canto superior esquerdo da figura e o número de linhas da figura (**NL**).

O primeiro **FOR ... NEXT** da linha 70 cuida de desenhar **NL** linhas. O programa tem que saber qual o número do bloco inicial de cada linha da figura *no banco de memória* (**BI**), em que coluna da tela ela começa (**C**) e quantos blocos ela tem (**NB**). O segundo **FOR ... NEXT** da linha 70 cuida de desenhar **NB** blocos.



A linha 80 se dedica a calcular a posição exata da tela (**X**,**Y**) e o número do bloco no banco (**N**), para que o laço entre as linhas 90 e 110 possa desenhar um novo bloco.

Esse tipo de organização das linhas **DATA** só é possível devido à disposição dos blocos no banco — carregue o gerador e observe o banco. As linhas 1 000 a 1 080 contêm os dados do crocodilo, da cobra, das quatro árvores e do macaco.

A linha 1 010, por exemplo, encerra os dados da cobra. Ela nos informa que esta deve ser desenhada com seu canto superior esquerdo na linha 17 e tem três linhas de blocos. O primeiro bloco da primeira linha ocupa a posição 160 no banco e será colocado na coluna 4. Essa primeira linha tem sete blocos. Com um programa assim você poderá fazer seus próprios cenários.

O CENÁRIO COMPLETO

Ao ser executado o programa, a tela se enche de blocos gráficos e o cenário parece completo. Uma cobra, um macaco e um sol radiante foram acrescentados pelas novas linhas. (Menos no Apple, em que todas as linhas são novas.) Poderíamos ter incorporado mais elefantes, árvores ou crocodilos à cena, mas, seja como for, dispomos agora de todo um conjunto de novos blocos para usar em nossos desenhos.

O SOL EM ALTA RESOLUÇÃO

Note que, além dos blocos, foram usados comandos gráficos (não há nenhum motivo para não se fazer a combinação das duas técnicas, já que isso dá excelentes resultados).

O Spectrum e o Apple constroem seu sol desenhando vários raios bem próximos, enquanto o MSX e o TRS-Color usam o comando **PAINT**.

Podemos ainda alterar o desenho empregando outros comandos gráficos, desenhando mais colinas ou traçando retas que façam o chão parecer irregular ou rachado.

Já comentamos a necessidade de usar matemática tanto no Apple como no TK-2000 devido à falta de recursos gráficos do BASIC.

Quando se trabalha com blocos gráficos, as variações possíveis são quase infinitas. Com eles, pode-se não só modificar o número de árvores, cobras e macacos, como também formar novas figuras. Um exemplo óbvio em nosso caso seria mudar a cor da copa das árvores (para branco ou cinza) e desenhar nuvens ou arbustos.

Devemos, contudo, prestar atenção nas cores. Se quisermos, por exemplo, usar as copas como arbustos, teremos que colocá-las no topo das colinas já que ambas são verdes. No caso do MSX, podemos mudar o tom de verde; no Apple não haverá problemas, já que a colina não foi pintada.

ANIMAÇÃO

As vantagens de construir as figuras com blocos gráficos não terminam aí: podemos também convertê-las em figuras móveis, fazendo certas modificações. Para animar os bichos, poderíamos definir um ou dois blocos extras — construindo um novo corpo para o elefante, por exemplo, ou uma cauda em posição diferente para a cobra, ou mesmo dando ao macaco uma banana para comer.

No MSX, podemos usar sprites para animar as figuras. Já no Apple e no TK-2000 encontraremos algumas dificuldades, pois os blocos gráficos são um tanto lentos. A alternativa é lançar mão do comando **DRAW**, com a ressalva de que, com sua utilização, é muito difícil preservar ao mesmo tempo a forma e a cor de uma figura. As linhas **DATA** necessárias para um **DRAW** são completamente diferentes das linhas que os blocos gráficos empregam.

JUNTE BLOCOS PARA FORMAR FIGURAS

Se quiser empregar blocos gráficos para animar os desenhos, as figuras destas páginas lhe mostrarão como os vários caracteres compõem a cobra e o macaco (o artigo anterior mostra um desenho semelhante para o crocodilo). Esses desenhos não correspondem às figuras do Apple, mas podem ser usados para criar elementos móveis (**DRAW**); para isso, é preciso apelar para o editor de figuras desse micro.

Enquanto o crocodilo, a cobra e o macaco parecem ter seu uso restrito aos desenhos de selvas e zoológicos, as árvores, nuvens e arbustos podem aparecer em outros desenhos.

GRAVAÇÃO DE BLOCOS

Os programas considerados até aqui empregam um grande número de blocos gráficos a título de ilustração, e para fornecer blocos potencialmente úteis em outros desenhos.

Todavia, é possível fazer desenhos muito interessantes usando uma quantidade bem menor de blocos. O truque para isso consiste em utilizar diversas ve-

zes os mesmos blocos. Desse modo, podemos, por exemplo, desenhar uma manada inteira de elefantes, ou colocar um muro em nosso cenário, preenchendo uma grande extensão da tela com apenas dois tipos de blocos gráficos.

Entretanto, como a produção de blocos não apresenta problemas, a grande vantagem em utilizar poucos blocos está em economizar tempo (no trabalho de criação e digitação) e espaço de memória (que as longas linhas **DATA** deixam de ocupar. O Spectrum, em particular, gasta muita memória para guardar essas linhas **DATA**).

Ao empregarmos linhas **DATA** para armazenar os padrões dos blocos, estaremos ocupando o dobro do espaço necessário, já que esses padrões serão transferidos para uma outra área da memória assim que rodarmos o programa. Em compensação, podemos economizar memória, se gravarmos os bancos de blocos diretamente em fita cassete, apagando a seguir as linhas **DATA**.

Para gravar os padrões do banco de blocos, use:

```
SAVE "nome" CODE x,y
```

onde x é o endereço inicial do segmento de memória que queremos gravar, e y é o seu comprimento.

Devemos, portanto, saber o endereço inicial do banco de blocos, tanto para colocar nele os valores das linhas **DATA**, como para mudar o apontador de caracteres UDG.

O comprimento do bloco é de fácil dedução: normalmente, ele é o número de blocos multiplicado por 8. O programa visto linhas atrás usa oito bancos para guardar todos os blocos necessários ao desenho da selva. Cada um desses bancos começa 256 bytes acima do banco anterior (ao contrário do número mínimo de 168 bytes), a fim de permitir que o apontador seja modificado com apenas um **POKE**, em vez de dois. Isso também significa que, se quisermos gravar essa porção de memória, devemos informar que ela tem 256 x 8 bytes de comprimento.

Grave os blocos da floresta com:

```
SAVE "floresta" CODE 63448,2048
```

Para carregar os blocos de volta, digite:

```
LOAD "" CODE
```

Para colocá-los em outro lugar da memória, escreva o novo endereço inicial após **CODE**.

O MSX grava o banco de blocos empregando o seguinte comando:

```
BSAVE "CAS:nome" x,y
```

onde x é o endereço inicial e y, o comprimento da parte da memória que queremos gravar.

No caso do programa acima, use:

```
BSAVE "CAS:SELVA" &HD100,&H1000
```

Isso grava apenas o padrão dos blocos. Lembramos que ficam faltando os bytes de cor. (Os blocos não serão visíveis pelo gerador.)

Para carregar os bytes da fita, use:

```
BLOAD "CAS:"
```

Para gravar os blocos guardados na página 2 de vídeo, devemos entrar no monitor com:

```
CALL - 151
```

Posicione a fita e digite:

```
4000.5000 W
```

Para carregar o banco, ainda no modo monitor:

```
4000.5000 R
```

No TK-2000 o processo se assemelha ao usado no Apple, só que devemos entrar no monitor com **LM**, e digitar:

```
A000.B000 W "floresta"
```

```
A000.B000 R ""
```

(para gravar e ler a fita, respectivamente).

Esse computador usa comandos **DRAW** em vez de linhas **DATA**, de forma que não há vantagem em gravar os blocos.

UM EDITOR DE TEXTOS (1)

Datilografe textos sem gastar dinheiro com programas sofisticados e sem quebrar a cabeça com processadores complicados de palavras, executando o programa deste artigo.

O processador de textos é uma das mais úteis aplicações para qualquer computador pessoal. Em certos casos, quando se quer melhorar os resultados, pode-se comprar um programa sofisticado em linguagem de máquina. Programas como esse, porém, são caros e o uso do editor nem sempre é simples.

A parte do programa coberta por este artigo constitui um padrão para um editor de textos simples, que está estruturado de forma a ser útil até mesmo àqueles que nunca viram um programa desse tipo. Com ele é possível criar memorandos, cartas ou quaisquer outros textos a serem enviados para a impressora. O texto pode ser gravado em disco ou em fita e carregado para uma posterior utilização.

Devido ao tamanho do programa completo — pelo menos 7,5k — a listagem foi dividida em três unidades, que não funcionarão adequadamente até que a última parte seja digitada. Apesar disso, será possível criar arquivos de texto no fim da segunda parte. A terceira parte é formada pela rotina para a impressora, que possibilita o envio de dados para impressão. Ordenação e rotina de busca estarão disponíveis.

Vejamos como o programa funciona, antes que sua primeira parte seja digitada. Ele está destinado a várias aplicações domésticas; mas, como em qualquer outro programa desse tipo, seus menus e telas de apresentação podem ser modificados ao bel-prazer do usuário. Todavia, se você resolver modificá-lo, tome cuidado para não alterar a numeração das linhas, de maneira que o computador possa acessá-las convenientemente, sobretudo se houver novos módulos a serem adicionados.

O MENU PRINCIPAL

Ao ser executada, a versão completa do programa apresentará imediatamente algumas opções para entrada e saída de dados. Isso prepara o sistema para o uso de cassete ou acionador de discos. No caso do Spectrum, quem possuir um Microdrive pode adaptar o programa para utilizá-lo de acordo com as indicações oferecidas.

Se for de seu interesse, você pode dar entrada por um meio e saída por outro e mudar mais tarde.

O menu principal é mostrado e as seguintes opções são apresentadas:

[C] ARREGAR
[G] RAVAR
[M] UDAR E/S
[E] DITAR
[L] IMPAR MEMÓRIA
[I] MPRIMIR
[S] AIR

CARREGAR

Se você quer continuar a trabalhar em um texto já existente, pressione **C** para iniciar a rotina que carrega o texto na memória. Qualquer outro aí presente será apagado. Portanto, uma mensagem "Confirme **(S/N)**" será inserida antes de se dar prosseguimento à rotina. Tecle **S** para continuar (ou **N** para interromper). Um nome de arquivo será então solicitado: ele é necessário para que se possa continuar. Em seguida, o referido arquivo é carregado na memória e pode ser trabalhado.

GRAVAR

A rotina de gravação é usada para criar arquivos seqüenciais para armazenamento dos textos. Quando essa opção for selecionada, teclando-se **G** — e desde que haja alguma coisa na memória — , será solicitado um nome para o arquivo. O Sinclair Spectrum gravará, mesmo que não haja nada na memória. Uma entrada nula não será aceita. O sistema já "sabe" qual opção de **E/S** está sendo usada e continua a pedir informações de acordo com as necessidades do cassete ou do acionador de discos. No final, pressione **<RETURN>** ou **<ENTER>** para efetuar a gravação.

Os usuários de disquete devem tomar bastante cuidado ao escolherem o nome do arquivo. Se o título de um arquivo já existente for usado, este será apagado, ficando no disco apenas o arquivo mais novo. Rotinas adicionais de gravação podem ser incorporadas a fim de proteger arquivos, trocar nomes etc.

ENTRADA/SAÍDA (E/S)

Se você quiser mudar os parâmetros iniciais de entrada e saída (em inglês, abreviados por I/O de *input/output*), pode fazê-lo a qualquer momento, teclando **M**. Digite sua opção. Isso configura automaticamente o programa com as novas necessidades.

O programa do Sinclair Spectrum funciona com fita ou Microdrive. As mudanças necessárias para o Microdrive aparecem após a listagem principal.

- ANALISE O MENU
- COMO CARREGAR E GRAVAR
- NOMES DE ARQUIVOS
- O TRABALHO DE EDIÇÃO
- COMO AJUSTAR CORES
- PARÂMETROS DE ENTRADA E SAÍDA
- TRABALHE COM O TEXTO
- COMO LIMPAR A MEMÓRIA
- PROGRAMA EDITOR DE TEXTOS

MODO DE EDIÇÃO

Pressionar a tecla **E** leva você a um menu secundário que mostra:

[T] OPO
[F] IM
[P] RÓXIMA LINHA
[C] OR (não para o Apple e o TRS-Color)
[M] ENU

As duas primeiras opções o levam ao início (**TOPO**) ou ao fim (**FIM**) do texto que está na memória. A terceira opção conduz para a última linha com que se estava trabalhando. A última o faz retornar ao menu principal.

A opção **COR** permite ajustar as cores de frente, fundo e bordas, de acordo com a sua preferência.

Cada uma das três opções iniciais coloca o programa no modo de entrada. A tela mostra linhas demarcando o início e o fim do texto (pode aparecer apenas uma, se já houver algum texto armazenado na memória). Na parte mais baixa da tela há uma "área de trabalho" e um indicador de "memória livre" (não disponível no micro Spectrum) que permite saber quantos caracteres disponíveis ainda se tem.

O texto é digitado na parte inferior da tela (área de trabalho). Os comandos **ENTER** ou **RETURN** o transferem para a memória e para a área de texto. O mesmo acontece quando duas linhas da tela são completadas.

Várias rotinas de edição estão embutidas no programa. Elas (e as teclas que as controlam) variam um pouco de máquina para máquina e serão discutidas em maior detalhe no momento oportuno. Toda a edição é feita na área de trabalho. Os controles de edição permitem que se movimente pela linha de texto na área de trabalho para inserção ou deleção (supressão) de caracteres, antes que o texto seja transferido para a memória. O texto que já estiver na memória — aquele que é mostrado no painel superior (área de texto) — deverá ser copiado, uma linha de cada vez, para a área de trabalho, antes de ser editado.

Novas linhas podem ser inseridas em qualquer ponto do texto. Isso se consegue pressionando a tecla apropriada para o modo "editor" (veja instruções detalhadas) e posicionando-se o marcador abaixo do ponto onde se quer inserir a nova linha.

Da mesma forma, o texto pode ser removido, uma linha de cada vez, selecionando-se o modo editor e posicionando-se o marcador abaixo da linha a ser apagada antes de se pressionar a tecla para deleção (supressão).

Teclas de controle permitem passear pelo texto, para a frente ou para trás, dez linhas de cada vez (o Spectrum é uma exceção). Assim, o texto pode ser visualizado antes da impressão ou de realizar alguma alteração.

Pode-se voltar ao menu do modo de edição pressionando-se a tecla de "escape" designada. É possível também voltar do modo de edição para o menu principal sempre que seja necessário alterar algum parâmetro do sistema.

LIMPEZA DA MEMÓRIA

A memória é limpa por intermédio de outra opção disponível a partir do menu principal. Mas esta só será oferecida depois que você responder sim à pergunta — "Confirma? (S/N)" — que se segue à teclagem de **L**. Note que essa op-

ção não redefine as opções de entrada e saída nem os parâmetros da impressora. Se você quiser reiniciar tudo, pressione **S** para retornar ao BASIC e, a seguir, **RUN**.

```
 10 POKE 23659,3
 20 BORDER 7: PAPER 7: INK 9:
  CLS
 30 DIM l(8): FOR N=1 TO 8:
 READ l(n): NEXT n
 40 LET Z$=""
100 LET ext=200: DIM t$(ext,32
): LET l1=32: LET pl=32
105 LET s$="
                "
110 LET t$(1)="ARQUIVO DO TOPO
DO TEXTO": LET t$(2)="--------
------------------------": LET
t$(3)=s$
120 LET t$(4)=s$: LET t$(5)="-
--------------------------
-": LET t$(6)="ARQUIVO DO FIM
 DO TEXTO"
130 LET t=1: LET b=6: LET p=4
140 CLS : PRINT  INVERSE 1;AT
0,8;" MENU PRINCIPAL "
150 PRINT AT 4,8;"1- Carregar
texto";AT 6,8;"2- Gravar texto
";AT 8,8;"3- Trocar papel";AT
10,8;"4- Editar";AT 12,8;"5- L
impar texto";AT 14,8;"6- Impri
mir texto";AT 16,8;"7- Alterar
impressora";AT 18,8;"8- Saida"
160 PRINT #1;TAB 7;"Selecione
opcao (1-8)"
170 LET a$=INKEY$: IF a$=""
THEN  GOTO 170
180 IF a$<"1" OR a$>"8" THEN
GOTO 170
190 LET a=VAL a$: CLS
200 GOSUB  l(a)
210 GOTO 140
500 CLS : PRINT  INVERSE 1;AT
4,8;" MENU - EDITOR "
510 PRINT AT 8,6;"1- Topo do t
exto";AT 10,6;"2- Fim do texto
";AT 12,6;"3- Proxima linha";
AT 14,6;"4- Sair do menu de ed
icao"
520 PRINT AT 18,7;"Selecione o
pcao (1-4)"
530 LET a$=INKEY$: IF a$=""
THEN  GOTO 530
540 IF a$<"1" OR a$>"4" THEN
GOTO 530
 550 LET a=VAL a$: CLS
 560 IF a=4 THEN  RETURN
 570 IF a=1 THEN  LET p=4
 580 IF a=2 THEN  LET p=b-2
 590 GOSUB 1000: GOSUB 2000
 600 GOTO 500
 900 PRINT AT 10,6;"Voce tem ce
 rteza?": PAUSE 0
 910 IF INKEY$="s" THEN  RUN
 920 RETURN
4000 RETURN
6000 REM carregar
6010 INPUT "Introduza o nome do
 arquivo", LINE n$: LOAD n$ DAT
A t$()
6020 LET b=VAL t$(1): LET t$(1)
="ARQUIVO DO TOPO DO TEXTO": RE
TURN
6200 REM gravar
6210 LET t$(1)=STR$ b
6220 INPUT "Introduza o nome do
 arquivo", LINE n$: IF n$="" OR
 LEN n$>10 THEN  GOTO 6220
6230 SAVE n$ DATA t$(): GOTO 60
20
6500 INPUT AT 0,0;"Introduza a
largura do  formula-rio (1-80)"
,pl: IF pl<1 OR pl>80 THEN  GOT
O 6500
6510 INPUT AT 0,0;"Introduza o
numero de caracterespor linha
(1-";(pl);")";l1: IF l1<1 OR l1
>pl THEN  GOTO 6510
6520 LET l1=l1+1: RETURN
9000 DATA 6000,6200,3000,500,90
0,4000,6500,9999
```

```
10 CLS:POKE 150,1
20 PMODE 0:PCLEAR 1:CLEAR 15500
30 DIM TX$(500)
40 BL$=CHR$(128):TL=1:CP=1:MW=8
0:TW=60:PL=66:TH=60:GP=10:LF$=S
TRING$(3,13):GOSUB 5000
50 TX$(0)=STRING$(32,195)
60 TX$(TL)=STRING$(32,188)
70 CLS:PRINT @9,BL$;"menu";BL$;
"principal";BL$:PRINT @104,"(C)
ARREGAR":PRINT @136,"(G)RAVAR":
PRINT @168,"(M)UDAR ENTRADA/SAI
DA"
80 PRINT @200,"(E)DITAR TEXTO":
PRINT @232,"(L)IMPAR MEMORIA":P
RINT @264,"(I)MPRIMIR":PRINT @2
96,"(A)LTERAR IMPRESSORA":PRINT
 @328,"(S)AIR";
90 B$=INKEY$:IF B$="" THEN 90
100 B=INSTR("CGMELIAS",B$):IF B
=0 THEN 90
110 ON B GOSUB 4500,4000,5000,1
000,160,3000,5500,130
120 GOTO 50
130 CLS:PRINT"VOCE TEM CERTEZA
(S/N)?"
140 R$=INKEY$:IF R$<>"S" AND R$
<>"N" THEN 140
150 IF R$="S" THEN CLS:END ELSE
 RETURN
160 CLS:PRINT@8,BL$;"limpar";BL
S;"a";BL$;"memoria";BL$:PRINT:P
RINT"VOCE TEM CERTEZA? (S/N)"
170 B$=INKEY$:IF B$<>"N" AND B$
<>"S" THEN 170
180 IF B$="N" THEN RETURN
```

```
190 FOR K=1 TO TL:TX$(K)="":NEX
T:TL=1:CP=1:RETURN
1000 CLS:PRINT @42,BL$"menu"BL$
"de"BL$"edicao"BL$:PRINT @104,"
tOPO DO TEXTO":PRINT @168,"fIM
DO TEXTO":PRINT @232,"pROXIMA L
INHA":PRINT @296,"mENU PRINCIPA
L"
1010 B$=INKEY$:IF B$="" THEN 10
10
1020 B=INSTR("TFPM",B$):IF B=0
THEN 1010
1030 ON B GOTO 1050,1060,1070,1
080
1050 CP=1:GOTO 1070
1060 CP=TL
1070 GOSUB 2090:GOSUB 1500:GOTO
 1000
1080 RETURN
1500 A$=" "
1510 P=0:PRINT @384,A$
1520 CH=PEEK(1408+P):T$=INKEY$:
IF T$="" THEN CH=(CH+64) AND 12
7:POKE 1408+P,CH:CH=(CH+64) AND
 127:POKE 1408+P,CH:GOTO 1520
1530 IF LEN(A$)=65 OR T$=CHR$(1
3) GOSUB 2000
1540 IF T$="^" THEN SF=0:GOSUB
2500:GOTO 1510
1550 IF P<LEN(A$)-1 AND T$=CHR$
(10) THEN A$=LEFT$(A$,P)+MID$(A
$,P+2):GOTO 1600 ELSE IF T$=CHR
$(10) THEN 1520
1560 IF T$=CHR$(12) THEN RETURN
1570 IF T$=CHR$(21) THEN P=-(LE
N(A$)-1)*(P=0):GOTO 1600
1580 IF T$<>CHR$(8) AND T$<>CHR
$(9) AND ASC(T$)<32 THEN 1510
1590 IF T$<>"" AND T$<>CHR$(8)
AND T$<>CHR$(9) THEN A$=LEFT$(A
$,P)+T$+MID$(A$,P+1):P=P+1
1600 PRINT @384,A$
1610 IF T$=CHR$(9) AND P<LEN(A$
)-1 THEN P=P+1
1620 IF T$=CHR$(8) AND P>0 THEN
 P=P-1
1630 GOTO 1520
2000 X=1:IF LEN(A$)>33 THEN X=2
2010 FOR K=TL+X TO CP+X STEP -1
:TX$(K)=TX$(K-X):NEXT
2020 IF LEN(A$)>33 THEN TX$(CP)
=LEFT$(A$,32):TX$(CP+1)=MID$(A$
,33,LEN(A$)-33) ELSE TX$(CP)=A$
2030 FOR K=0 TO X-1
2040 IF RIGHT$(TX$(CP+K),1)=" "
 THEN TX$(CP+K)=LEFT$(TX$(CP+K)
,LEN(TX$(CP+K))-1):GOTO 2040
2050 NEXT
2060 A$=" ":P=0:PRINT @384,A$
2070 PRINT @416,""
2080 TL=TL+X:CP=CP+X
2090 IF CP<5 THEN ST=0 ELSE ST=
CP-5
2100 PRINT @0,;:FOR K=ST TO ST+
9:PRINT TX$(K);:IF LEN(TX$(K))<
32 THEN PRINT
2110 IF K=CP-1 THEN PRINT ">"
2120 NEXT:PRINT STRING$(32,140)
2130 PRINT @480,BL$;"mem";BL$;"
livre=";32*(501-TL);BL$;BL$;"cl
ear=menu";BL$;:POKE 1534,32:POK
E 1529,61:RETURN
```

```
10 CLS:KEY FF
20 COLOR 15,4,4:CLEAR 18000
30 DIM TX$(500):V$=STRING$(39,1
95)
40 TL=1:CP=1:MW=80:TW=60:PL=66:
TH=60:GP=10:L1$=CHR$(13):LF$=ST
RING$(3,L1$)
50 TX$(0)=STRING$(39,95):TX$(TL
)=TX$(0)
60 CLS:LOCATE 7:PRINT"M E N U
 P R I N C I P A L":LOCATE 9,4:
PRINT"[C]arregar":LOCATE 9,6:PR
INT"[G]ravar"
70 LOCATE 9,8:PRINT"[E]ditar te
xto":LOCATE 9,10:PRINT"[L]impar
 memória":LOCATE 9,12:PRINT"[I]
mprimir":LOCATE 9,14:PRINT"[A]l
terar impressora":LOCATE 9,16:P
RINT"[S]air"
80 LOCATE 13,20:PRINT"Opção ?";
90 B$=INKEY$:IF B$=""THEN 90
100 B=INSTR("CGELIAS",B$):IF B=
0 THEN 90
110 ON B GOSUB 4500,4000,1000,1
70,3000,5500,130
120 GOTO 50
130 CLS:BL$="Fim de programa":G
OSUB 220
140 LOCATE 12,10:PRINT"Confirme
 (S/N) ";
150 R$=INKEY$:IF R$="" THEN 150
160 IF R$="S" THEN CLS:END ELSE
 RETURN
170 CLS:BL$="Limpa a memória":G
OSUB 220
180 LOCATE 12,10:PRINT"Confirme
 (S/N) ";
190 R$=INKEY$:IF R$="" THEN 190
200 IF R$<>"S" THEN RETURN
210 FOR K=1 TO TL:TX$(K)="":NEX
T:TL=1:CP=1:RETURN
220 PRINTBL$:PRINT V$:RETURN
1000 CLS:BL$="Menu de edição":G
OSUB 220:LOCATE 9,8:PRINT"[T]op
o do texto":LOCATE 9,10:PRINT"[
F]im do texto":LOCATE 9,12:PRIN
T"[P]róxima linha":LOCATE 9,14:
PRINT"[M]enu principal"
1010 LOCATE 13,17:PRINT"Opção ?
";
1020 B$=INKEY$:IF B$="" THEN 10
20
1030 B=INSTR("TFPM",B$):IF B=0
THEN 1020
1040 ON B GOTO 1050,1060,1070,1
080
1050 CP=1:GOTO 1070
1060 CP=TL
1070 CLS:GOSUB 2080:GOSUB 1090:
GOTO 1000
1080 RETURN
1090 A$=" "
1100 P=0:LOCATE 0,16:PRINT A$;S
PC(80)
1110 LOCATE P+39*(P>38),16-(P>3
8):PRINTCHR$(219);
1120 T$=INKEY$:IF T$="" THEN 11
20
1130 IF LEN(A$)=79 OR T$=CHR$(1
3) THEN GOSUB 2000
1140 IF T$=CHR$(5) THEN SF=0:GO
SUB 2500:GOTO 1100
1150 IF P<LEN(A$)-1 AND T$=CHR$
(4) THEN A$=LEFT$(A$,P)+MID$(A$
,P+2):GOTO 1200
1160 IF T$=CHR$(27) THEN RETURN
1170 IF T$=CHR$(1) THEN P=0:GOT
O 1200 ELSE IF T$=CHR$(19) THEN
 P=LEN(A$)-1:GOTO 1200
1180 IF T$<>CHR$(28) AND T$<>CH
R$(29) AND T$<CHR$(32) THEN 111
0
1190 IF T$<>"" AND T$<>CHR$(28)
 AND T$<>CHR$(29) THEN A$=LEFT$
(A$,P)+T$+MID$(A$,P+1):P=P+1
1200 LOCATE 0,16:PRINTA$;SPC(80
)
1210 IF T$=CHR$(28) AND P<LEN(A
$)-1 THEN P=P+1
1220 IF T$=CHR$(29) AND P>0 THE
N P=P-1
1230 GOTO 1110
2000 X=1:IF LEN(A$)>40 THEN X=2
2010 FOR K=TL+X TO CP+X STEP-1:
TX$(K)=TX$(K-X):NEXT
2020 IF LEN(A$)>40 THEN TX$(CP)
=LEFT$(A$,39):TX$(CP+1)=MID$(A$
,40,LEN(A$)-40) ELSE TX$(CP)=A$
2030 FOR K=0 TO X-1
2040 IF RIGHT$(TX$(CP+K),1)=" "
```

```
 THEN TX$(CP+K)=LEFT$(TX$(CP+K)
,LEN(TX$(CP+K))-1):GOTO 2040
2050 NEXT
2060 A$=" ":P=0:LOCATE 0,16:PRI
NTA$;SPC(80)
2070 TL=TL+X:CP=CP+X
2080 IF CP<9 THEN ST=0 ELSE ST=
CP-9
2090 LOCATE 0,0:FOR K= ST TO ST
+9:PRINTTX$(K);SPC(39-LEN(TX$(K
)))
2100 IF K=CP-1 THEN PRINTCHR$(1
75);SPC(38)
2110 NEXT:PRINTV$
2120 LOCATE 0,23:PRINT"Mem livr
e = ";39*(501-TL);"  ";TAB(26);
"<ESC>=Menu";:RETURN
```

```
10  HOME
20  DIM TX$(500)
30 TL = 1:CP = 1:MW = 80:TW = 6
0:PL = 66:TH = 60:GP = 10:LF$ =
  CHR$ (13) +  CHR$ (13) +  CHR
$ (13)
40 L1 = 6:S1 = L1:L2 = 1:S2 = L
2:D$ =  CHR$ (4)
50  FOR Z = 1 TO 40:TX$(0) = TX
$(0) + "-":A$ = A$ + "*": NEX
T
60  GOSUB 5000
70 TX$(TL) = TX$(0)
80  HOME : HTAB 8: PRINT "M E N
 U   P R I N C I P A L"
90  VTAB 5: HTAB 10: PRINT "[C]
arregar": PRINT : HTAB 10: PRIN
T "[G]ravar": PRINT : HTAB 10:
PRINT "[M]udar E/S"
100  PRINT : HTAB 10: PRINT "[E
]ditar texto": PRINT : HTAB 10:
 PRINT "[L]impar memoria": PRIN
T : HTAB 10: PRINT "[I]mprimir"
: PRINT : HTAB 10: PRINT "[A]lt
erar impressora": PRINT : HTAB
10: PRINT "[S]air"
110  PRINT : PRINT : PRINT  TAB
( 13)"Opcao =>";
120  GET B$: IF B$ =  CHR$ (13)
 THEN 120
130 BB$ = "CGMELIAS":B = 0: FOR
 Z = 1 TO 8: IF B$ =  MID$ (BB$
,Z,1) THEN B = Z
140  NEXT : IF B = 0 THEN 120
150  ON B GOSUB 4500,4000,5000,
1000,210,3000,5500,170
160  GOTO 70
170  HOME :BL$ = "FIM DE PROGRA
MA": GOSUB 250
180  HTAB 12: VTAB 10: PRINT "C
ONFIRME (S/N)";: GET R$
190  IF R$ <  > "S" THEN  RETUR
N
200  HOME : END
210  HOME :BL$ = "LIMPAR A MEMO
RIA": GOSUB 250
220  HTAB 12: VTAB 10: PRINT "C
ONFIRME (S/N)";: GET R$
230  IF R$ <  > "S" THEN  RETUR
N
240  FOR K = 1 TO TL:TX$(K) = "
": NEXT :TL = 1:CP = 1: RETURN
250  INVERSE : PRINT BL$;: PRIN
T  SPC( 40 -  LEN (BL$)): NORMA
L : RETURN
1000  HOME : PRINT  TAB( 12)"ME
NU DE EDICAO"
1010  VTAB 9: HTAB 10: PRINT "[
T]OPO DO TEXTO"
1020  PRINT : PRINT  TAB( 10)"[
F]IM DO TEXTO"
1030  PRINT : PRINT  TAB( 10)"[
P]ROXIMA LINHA"
1040  PRINT : PRINT  TAB( 10)"[
M]ENU PRINCIPAL"
1050  PRINT : PRINT : PRINT  TA
B( 13)"Opcao =>";
1060  GET B$: IF B$ =  CHR$ (13
) THEN 1060
1070 BB$ = "TFPM":B = 0: FOR Z
= 1 TO 4: IF B$ =  MID$ (BB$,Z,
1) THEN B = Z
1080  NEXT : IF B = 0 THEN 1060

1090  ON B GOTO 1110,1120,1130,
1140
1110 CP = 1: GOTO 1130
1120 CP = TL
1130  HOME : GOSUB 2090: GOSUB
1500: GOTO 1000
1140  RETURN
1500 A$ = "  "
1510 P = 1: HTAB 1: VTAB 20: CA
LL  - 958: PRINT  MID$ (A$,2);
1520  HTAB P: VTAB 20 + (P > 40
): GET T$
1530  IF  LEN (A$) = 82 OR T$ =
  CHR$ (13) THEN  GOSUB 2000
1540  IF T$ =  CHR$ (5) THEN  G
OSUB 2500: GOTO 1510
1550  IF T$ =  CHR$ (27) THEN
RETURN
1560  IF T$ =  CHR$ (1) THEN P
= 1: GOTO 1620
1570  IF T$ =  CHR$ (19) THEN P
 =  LEN (A$) - 1: GOTO 1620
1580  IF T$ =  CHR$ (4) AND (P
<  LEN (A$) - 1) THEN A$ =  LEF
T$ (A$,P) +  MID$ (A$,P + 2): G
OTO 1620
1590  IF T$ =  CHR$ (4) THEN 15
20
1600  IF T$ <  > CHR$ (8) AND
T$ <  > CHR$ (21) AND T$ <  CH
R$ (32) THEN 1520
1610  IF T$ <  > CHR$ (8) AND
T$ <  > CHR$ (21) THEN A$ =  L
EFT$ (A$,P) + T$ +  MID$ (A$,P
+ 1):P = P + 1
1620  HTAB 1: VTAB 20: CALL  -
958: PRINT  MID$ (A$,2);
1630  IF T$ =  CHR$ (8) AND (P
> 1) THEN P = P - 1
1640  IF T$ =  CHR$ (21) AND (P
 <  LEN (A$) - 1) THEN P = P +
1
1650  GOTO 1520
2000 X = 1:A$ =  MID$ (A$,2): I
F  LEN (A$) > 41 THEN X = 2
2010  FOR K = TL + X TO CP + X
STEP  - 1:TX$(K) = TX$(K - X):
NEXT
2020  IF  LEN (A$) > 41 THEN TX
$(CP) =  LEFT$ (A$,40):TX$(CP +
 1) =  MID$ (A$,41): GOTO 2040
2030 TX$(CP) = A$: IF A$ = " "
THEN 2070
2040  FOR K = 0 TO X - 1
2050  IF  RIGHT$ (TX$(CP + K),1
) = " " THEN TX$(CP + K) =  LEF
T$ (TX$(CP + K), LEN (TX$(CP +
K)) - 1): GOTO 2050
2060  NEXT
2070 A$ = "  ":P = 1: HTAB 1: V
TAB 20: CALL  - 958: PRINT  MID
$ (A$,2);
2080 TL = TL + X:CP = CP + X
2090  IF CP < 9 THEN ST = 0: GO
TO 2110
2100 ST = CP - 9
2110  HTAB 1: VTAB 1: FOR K = S
T TO ST + 13: PRINT TX$(K);: IF
  LEN (TX$(K)) < 40 THEN  CALL
 - 868: PRINT
2120  IF K = CP - 1 THEN  CALL
 - 868: PRINT ">"
2130  NEXT : PRINT AS$;: PRINT
"memoria livre=";40 * (501 - TL
)
2140  RETURN
```

Quais são as principais diferenças entre o editor de textos de *INPUT* e os editores vendidos nas lojas especializadas?

A principal diferença consiste em que nosso editor é programado em BASIC, e os editores comerciais, em código de máquina. Tais editores atuam sobre uma "janela de texto", ou seja, permitem que se edite qualquer segmento de texto, acrescentando ou eliminando palavras, frases e parágrafos, por meio de simples movimentos do cursor na tela.

Em contraste com isso, nosso programa edita linha por linha do texto, exigindo que as linhas sejam escritas na parte inferior da tela e transferidas para o campo superior.

Esse modo de organizar o texto, junto com a lentidão inerente à linguagem BASIC, é a causa de outra diferença fundamental: o tempo de gravação e leitura dos textos em editores comerciais é muito menor.

Outro atributo dos editores comerciais é a hifenação automática das palavras. Somada ao alinhamento à direita, essa característica dá ao texto impresso um aspecto mais natural e profissional. Não há nenhum motivo, contudo, para que uma função de hifenação não possa ser incluída em nosso editor.

CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR LINHA

LINHA	FABRICANTE	MODELO
Apple II +	Appletronica	Thor 2010
Apple II +	CCE	MC-4000 Exato
Apple II +	CPA	Absolutus
Apple II +	CPA	Polaris
Apple II +	Digitus	DGT-AP
Apple II +	Dismac	D-8100
Apple II +	ENIAC	ENIAC II
Apple II +	Franklin	Franklin
Apple II +	Houston	Houston AP
Apple II +	Magnex	DM II
Apple II +	Maxitronica	MX-2001
Apple II +	Maxitronica	MX-48
Apple II +	Maxitronica	MX-64
Apple II +	Maxitronica	Maxitronic I
Apple II +	Microcraft	Craf II Plus
Apple II +	Milmar	Apple II Plus
Apple II +	Milmar	Apple Master
Apple II +	Milmar	Apple Senior
Apple II +	Omega	MC-400
Apple II +	Polymax	Maxxi
Apple II +	Polymax	Poly Plus
Apple II +	Spectrum	Microengenho I
Apple II +	Spectrum	Spectrum ed
Apple II +	Suporte	Venus II
Apple II +	Sycomig	SIC I
Apple II +	Unitron	AP II
Apple II +	Victor do Brasil	Elppa II Plus
Apple II +	Victor do Brasil	Elppa Jr.
Apple IIe	Microcraft	Craft IIe
Apple IIe	Microdigital	TK-3000 IIe
Apple IIe	Spectrum	Microengenho II
MSX	Gradiente	Expert GPC-1
MSX	Sharp	Hotbit HB-8000
Sinclair Spectrum	Microdigital	TK-90X
Sinclair Spectrum	Timex	Timex 2000
Sinclair ZX-81	Apply	Apply 300
Sinclair ZX-81	Engebras	AS-1000
Sinclair ZX-81	Filcres	NEZ-8000
Sinclair ZX-81	Microdigital	TK-82C
Sinclair ZX-81	Microdigital	TK-83
Sinclair ZX-81	Microdigital	TK-85
Sinclair ZX-81	Prologica	CP-200
Sinclair ZX-81	Ritas	Ringo R-470
Sinclair ZX-81	Timex	Timex 1000
Sinclair ZX-81	Timex	Timex 1500
TRS-80 Mod. I	Dismac	D-8000
TRS-80 Mod. I	Dismac	D-8001/2
TRS-80 Mod. I	LNW	LNW-80
TRS-80 Mod. I	Video Genie	Video Genie I
TRS-80 Mod.III	Digitus	DGT-100
TRS-80 Mod.III	Digitus	DGT-1000
TRS-80 Mod.III	Kemitron	Naja 800
TRS-80 Mod.III	Prologica	CP-300
TRS-80 Mod.III	Prologica	CP-500
TRS-80 Mod.III	Sysdata	Sysdata III
TRS-80 Mod.III	Sysdata	Sysdata Jr.
TRS-80 Mod.IV	Multix	MX-Compacto
TRS-80 Mod.IV	Sysdata	Sysdata IV
TRS-Color	Codimex	CS-6508
TRS-Color	Dynacom	MX-1600
TRS-Color	LZ	Color 64
TRS-Color	Microdigital	TKS-800
TRS-Color	Prologica	CP-400

CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR NOME

FABRICANTE	MODELO	PAÍS	LINHA
Appletronica	Thor 2010	Brasil	Apple II +
Apply	Apply 300	Brasil	Sinclair ZX-81
CCE	MC-4000 Exato	Brasil	Apple II +
CPA	Absolutus	Brasil	Apple II +
CPA	Polaris	Brasil	Apple II +
Codimex	CS-6508	Brasil	TRS-Color
Digitus	DGT-100	Brasil	TRS-80 Mod.III
Digitus	DGT-1000	Brasil	TRS-80 Mod.III
Digitus	DGT-AP	Brasil	Apple II +
Dismac	D-8000	Brasil	TRS-80 Mod. I
Dismac	D-8001/2	Brasil	TRS-80 Mod. I
Dismac	D-8100	Brasil	Apple II +
Dynacom	MX-1600	Brasil	TRS-Color
ENIAC	ENIAC II	Brasil	Apple II +
Engebras	AS-1000	Brasil	Sinclair ZX-81
Filcres	NEZ-8000	Brasil	Sinclair ZX-81
Franklin	Franklin	USA	Apple II +
Gradiente	Expert GPC1	Brasil	MSX
Houston	Houston AP	Brasil	Apple II +
Kemitron	Naja 800	Brasil	TRS-80 Mod.III
LNW	LNW-80	USA	TRS-80 Mod. I
LZ	Color 64	Brasil	TRS-Color
Magnex	DM II	Brasil	Apple II +
Maxitronica	MX-2001	Brasil	Apple II +
Maxitronica	MX-48	Brasil	Apple II +
Maxitronica	MX-64	Brasil	Apple II +
Maxitronica	Maxitronic I	Brasil	Apple II +
Microcraft	Craft II Plus	Brasil	Apple II +
Microcraft	Caft IIe	Brasil	Apple IIe
Microdigital	TK-3000 IIe	Brasil	Apple IIe
Microdigital	TK-82C	Brasil	Sinclair ZX-81
Microdigital	TK-83	Brasil	Sinclair ZX-81
Microdigital	TK-85	Brasil	Sinclair ZX-81
Microdigital	TK-90X	Brasil	Sinclair Spectrum
Microdigital	TKS-800	Brasil	TRS-Color
Milmar	Apple II Plus	Brasil	Apple II +
Milmar	Apple Master	Brasil	Apple II +
Milmar	Apple Senior	Brasil	Apple II +
Multix	MX-Compacto	Brasil	TRS-80 Mod.IV
Omega	MC-400	Brasil	Apple II +
Polymax	Maxxi	Brasil	Apple II +
Polymax	Poly Plus	Brasil	Apple II +
Prologica	CP-200	Brasil	Sinclair ZX-81
Prologica	CP-300	Brasil	TRS-80 Mod.III
Prologica	CP-400	Brasil	TRS-Color
Prologica	CP-500	Brasil	TRS-80 Mod.III
Ritas	Ringo R-470	Brasil	Sinclair ZX-81
Sharp	Hotbit HB-8000	Brasil	MSX
Spectrum	Microengenho I	Brasil	Apple II +
Spectrum	Microengenho II	Brasil	Apple IIe
Spectrum	Spectrum ed	Brasil	Apple II +
Suporte	Venus II	Brasil	Apple II +
Sycomig	SIC I	Brasil	Apple II +
Sysdata	Sysdata III	Brasil	TRS-80 Mod.III
Sysdata	Sysdata IV	Brasil	TRS-80 Mod.IV
Sysdata	Sysdata Jr.	Brasil	TRS-80 Mod.III
Timex	Timex 1000	USA	Sinclair ZX-81
Timex	Timex 1500	USA	Sinclair ZX-81
Timex	Timex 2000	USA	Sinclair Spectrum
Unitron	AP II	Brasil	Apple II +
Victor do Brasil	Elppa II Plus	Brasil	Apple II +
Victor do Brasil	Elppa Jr.	Brasil	Apple II +
Video Genie	Video Genie I	USA	TRS-80 Mod. I

UM LOGOTIPO PARA CADA MODELO DE COMPUTADOR

INPUT foi especialmente projetado para microcomputadores compatíveis com as sete principais linhas existentes no mercado.
Os blocos de textos e listagens de programas aplicados apenas a determinadas linhas de micros podem ser identificados por meio dos seguintes símbolos:

Sinclair ZX-81

TRS-80

TK-2000

MSX

Spectrum

TRS-Color

Apple II

Quando o emblema for seguido de uma faixa, então tanto o texto como os programas que se seguem passam a ser específicos para a linha indicada.

NOVOS LANÇAMENTOS, NOVOS SUCESSOS.

JÁ NAS LIVRARIAS

A FRATERNIDADE DA PEDRA
David Morrell

Um grupo secreto, sob direção de um padre armado, passa a agir contra o terrorismo. Mas, será que violência se combate com mais violência? Eis o dilema de Drew, agente da Lei envolvido com fatos e figuras do mundo real, num livro surpreendente do criador de Rambo.

AVENTUREIROS E MILIONÁRIOS
Clark Howard

No romance do Texas, a saga da descoberta de um poço de petróleo e o drama de um casal que herda um lote de terra aparentemente sem valor e enfrenta com coragem os poderosos do lugar, até vencer. Uma vitória antes de tudo moral, numa história forte e envolvente, com realistas cenas de amor.

CAÇADA SEM FIM
Bryan Forbes

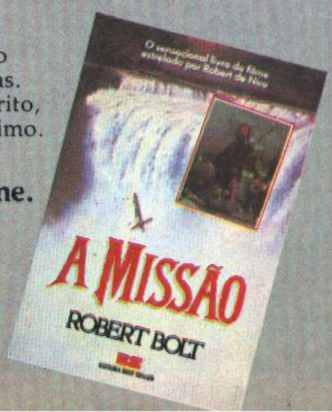
Uma brilhante história de espionagem envolvendo a KGB. Por que matar uma ex-espiã que já tinha sido desmascarada e torturada tempos atrás? Um agente inglês, seu antigo amante, enfrenta um desafio: descobrir por que ela foi morta... e por que *agora!*

PODER
Howard Fast

Um líder sindical com a volúpia do poder, a luta pelos direitos dos trabalhadores, nos Estados Unidos, e sua manipulação por corruptos e oportunistas; o jogo das ambições políticas. Admiravelmente escrito, um romance atualíssimo.

SUPERSEXO
Alexandra Penney

Se não for o primeiro, este vai ser o último e definitivo guia para o prazer que o leitor poderá seguir: um livro que derruba mitos, faz sugestões provocantes e propõe técnicas ousadas para se chegar ao supersexo, uma relação intensa e especial entre os casais, que não exclui o romantismo.



Não perca também: A MISSÃO, de Robert Bolt, o livro do filme.

EDITORA BEST SELLER